# GENERAL RELATIVISTIC SMOOTHED PARTICLE HYDRODYNAMICS: A MULTI-SCALE FORMULATION OF FLUID FLOW IN NUMERICAL RELATIVITY

A Thesis
Presented to
The Academic Faculty

by

Matthew Christopher Kinsey

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Physics

Georgia Institute of Technology
August 2016

# GENERAL RELATIVISTIC SMOOTHED PARTICLE HYDRODYNAMICS: A MULTI-SCALE FORMULATION OF FLUID FLOW IN NUMERICAL RELATIVITY

Approved by:

Professor John Wise,
Committee Chair
School of Physics
*Georgia Institute of Technology*

Professor Pablo Laguna, Advisor
School of Physics
*Georgia Institute of Technology*

Professor Tamara Bogdanovic
School of Physics
*Georgia Institute of Technology*

Professor Richard Vuduc
School of Computational Science and Engineering
*Georgia Institute of Technology*

Professor Deirdre Shoemaker
School of Physics
*Georgia Institute of Technology*

Date Approved: 20 July 2016

*To my grandmother,*

*Erma Kinsey,*

*and my late grandfather,*

*Raymond Kinsey,*

*for inspiring me from a young age to always keep learning and never give up on what I set out to do. Whether teaching me how an engine works or simply challenging me with puzzles and games, your wisdom has always been inspiring*[1].

[1] *I also really appreciated all the extra ice cream and late bed times. I still haven't told Mom and Dad.*

# ACKNOWLEDGEMENTS

First, I would like to thank my committee for their constructive input on this thesis. I would also like to thank all the people that helped guide me during the beginnings of my graduate studies, Tanja Bode, Roland Haas, Jim Healy, Roseanne Cheng, Deirdre Shoemaker, and, of course, my advisor Pablo Laguna.

Additionally, I would like to thank my fellow graduate students, Michael Clark, Jim Waters, Jeff Tithof and Brad Taylor, who have been true friends during my time here. Special thanks are owed to Katharine Tallaksen for proofreading my thesis more times than anyone else and teaching me how one properly uses commas,

Finally, I would like to thank my family: My parents, Dennis and Connie Kinsey have always been supportive and without whom I would quite literally not be here today. My wonderful sister Amy Kinsey, who is the slightly better version of myself in every conceivable way. Without you guys, I never would have been able to do this.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF SYMBOLS OR ABBREVIATIONS

| | |
|---|---|
| $\alpha$ | Lapse scalar. |
| $\bar{E}$ | GRSPH energy variable. |
| $\bar{U}^{\mu}$ | Fluid four velocity in the Eulerian frame. |
| $\beta^{\mu}$ | Shift vector. |
| $c_s$ | Speed of sound. |
| $D^{\star}$ | GRSPH density variable. |
| $\eta_{\mu\nu}$ | Flat spacetime metric, $\eta_{\mu\nu} = \mathrm{diag}(-1, 1, 1, 1)$. |
| $g_{\mu\nu}$ | Spacetime four-metric. |
| $g$ | Spacetime four-metric determinant. |
| $\gamma$ | Spatial metric determinant. |
| $\Gamma$ | Adiabatic index. |
| $\gamma_{ij}$ | Spatial metric of a hypersurface. |
| $h$ | SPH kernel smoothing length. |
| $J^{\mu}$ | Four current. |
| $m$ | Fluid rest mass. |
| $\mathcal{M}$ | Mach number. |
| $\nabla_{\mu}$ | Covariant derivative with respect to $x^{\mu}$. |
| $n^{\mu}$ | Unit vector normal to a hypersurface. |
| $\omega$ | Enthalpy. |
| $\Omega$ | Correction factor for variable smoothing lengths. |
| $P$ | Fluid Pressure. |
| $\partial_{\mu}$ | Partial derivative with respect to $x^{\mu}$. |
| $q$ | Artificial viscous pressure. |
| $\rho$ | Energy density. |
| $\rho_0$ | Rest mass energy density. |

| | |
|---|---|
| $T_{\mu\nu}$ | Stress-energy tensor. |
| $\tilde{\alpha}, \tilde{\beta}$ | Artificial viscous pressure parameters. |
| $u$ | Specific internal energy. |
| $U^{\mu}$ | Fluid four velocity it in coordinate frame. |
| $\vec{G}$ | Integral based kernel derivative ($\nabla W(\vec{r}, h) \rightarrow \vec{G}$). |
| $v^{i}$ | Fluid three velocity in the coordinate frame. |
| $V^{i}$ | Fluid three velocity Eulerian frame. |
| $W$ | Fluid Lorentz factor. |
| $W_{ab}(h)$ | SPH smoothing kernel between particles $a$ and $b$ ($W(\lvert\vec{r}_a - \vec{r}_b\rvert, h)$). |
| $W(\vec{r}, h)$ | SPH smoothing kernel. |

# SUMMARY

With the recent first detection of gravitational waves, numerical relativity provides us with the most promising tools of astronomical discovery, particularly for strong dynamical gravity phenomena where analytic solutions remain elusive. However, finding numerical solutions to the Einstein field equations of General Relativity and their accompanying matter source equations often comes at a steep computational cost. In this thesis, I present a Lagrangian formalism for solving the equations of relativistic hydrodynamics in a dynamical 3+1 spacetime using 'smoothed particle hydrodynamics' (SPH) techniques. This method comes with numerous advantages over more traditional Eulerian methods. In particular, the resolution of SPH naturally follows the density distribution of the fluid: a distribution that may span many orders of magnitude in relevant astrophysical problems. The accuracy and validity of this method is then established by showing agreement with well-established analytical test cases in relativistic hydrodynamics. Additionally, I highlight the parallel properties of this method and discuss how this approach naturally lends itself well to a scientific computing environment that is increasingly seeing gains, not from higher clock rates, but rather a push towards massive parallelism.

# CHAPTER I

# INTRODUCTION

With the recent detection of gravitational waves by the Laser Interferometer Gravitational-Wave Observatory (LIGO) [2, 1], the field of relativistic astrophysics has entered a new era of observation. A prediction of Einstein's theory of general relativity, these waves represent a means to look at the universe in a brand new "light", providing new information to complement past observations using electromagnetic waves.

The progenitor systems for detectable gravitational waves represent some of the most studied systems in general relativity which lack analytical solutions. Due to the complexity of the problem, we are left to solve Einstein's field equations and the accompanying equations of relativistic hydrodynamics using numerical methods, often at a large computational cost. For this reason, numerical relativity (NR) has played a critical role in the discovery of gravitational waves by providing the most accurate template gravitational waveforms. Such waveforms are critical to performing the matched filtering that is required to find gravitational signals deeply embedded in the noise of the LIGO detectors. Moreover, the simulation of hydrodynamics in the vicinity of these compact objects will be able to provide even more information about the systems should a traditional instrument be able to detect a coincident electromagnetic or neutrino signal. With neutron star/neutron star (NSNS) and neutron star/black hole (NSBH) binaries expected to be future sources for detected gravitational wave signals, our ability to simulate relativistic hydrodynamics has become a primary interest of the field of numerical relativity.

In this work, we present a method for solving the equations of relativistic hydrodynamics, in strong dynamical gravity, using a particle method. This method, dubbed

"smoothed particle hydrodynamics" (SPH), comes with numerous advantages over more traditional Eulerian methods. In particular, the resolution of SPH follows the density distribution of the fluid: a distribution that may span many orders of magnitude in relevant astrophysical problems. As a result, the GRSPH method is well suited for many of the problems of interest in the spacetimes around compact objects like black holes and neutron stars, problems where the physical scales can span many orders of magnitude. Unlike Eulerian methods, the boundaries between the fluid and vacuum are modeled easily, with no need to introduce an artificial "atmosphere". By coupling such a method to a code that is capable of solving Einstein's equations of general relativity in the presence of dynamical masses, we will be able to model many of the systems of interest. For example, by using such a multiscale method, one can model a NSBH merger with great accuracy.

The GRSPH method has been previously applied to the case of irrotational NSNS mergers in conformally flat gravity [24] and BHNS mergers [55] with a fixed background spacetime and has shown great potential in accurately modeling these systems. Beyond the merger of compact objects, such a code could also be used to explore the tidal disruption of stars by supermassive black holes, or the mechanisms of accretion onto compact objects. More practically, SPH is a highly parallel method which naturally lends itself well to a scientific computing environment that is increasingly seeing gains, not from higher clock rates, but rather a push towards massive parallelism.

In this work, we have developed a formalism to solve the equations of general relativistic hydrodynamics in the presence of strong dynamical gravity. By casting the equations of relativistic hydrodynamics into the Lagrangian form, we enable them to be solved using the SPH method. We have also developed the computational framework necessary to use such a method at the computational scale of modern numerical relativity codes. In doing this, we hope to couple our method for relativistic hydrodynamics with a code capable of solving Einstein's equations for dynamical

systems.

# CHAPTER II

# NUMERICAL RELATIVITY BACKGROUND

In this chapter, we will review the foundations of General Relativity as they pertain to hydrodynamics in a curved spacetime. We will adopt the following conventions throughout this thesis [45]. We set both the speed of light, $c$, and Newton's gravitational constant, $G$, to unity ($c = G = 1$). We also adopt the metric signature $(-,+,+,+)$ such that the Minkowski metric of a flat spacetime is written as $\eta_{\mu\nu} = \text{diag}(-1,1,1,1)$.

## *2.1 The 3+1 Formalism*

We must start by casting the equations of General Relativity into a form suitable for numerical integration [8]. The most frequently used approach to split the spacetime is that of Arnowitt-Deser-Misner (ADM) [6]. Using the ADM method, one first chooses a scalar function, $t$, on the four dimensional spacetime with four-metric $g_{\mu\nu}$. By choosing $t$, one can then foliate the manifold into non-intersecting space-like hypersurfaces, $\Sigma_t$, by using the level sets of $t$, i.e. the $t$ = const hypersurfaces (See Figure 2.1).

Starting from one such hypersurface, the three-space can then be numerically integrated forward along a time-like vector $t^\mu$ (Equation 2.6). This method is commonly referred to as the 3+1 formalism. On this hypersurface, $\Sigma_t$, we define the one-form

$$\Omega_\mu = \nabla_\mu t \tag{2.1}$$

as a vector normal to the surface of constant $t$. We then normalize it by defining the "lapse" scalar as the inverse of its magnitude (i.e. $\alpha \equiv 1/||\Omega|| = 1/\sqrt{\Omega \cdot \Omega} = 1/\sqrt{\Omega_\mu \Omega^\mu} = 1/\sqrt{g^{\mu\nu}\Omega_\mu\Omega_\nu}$),

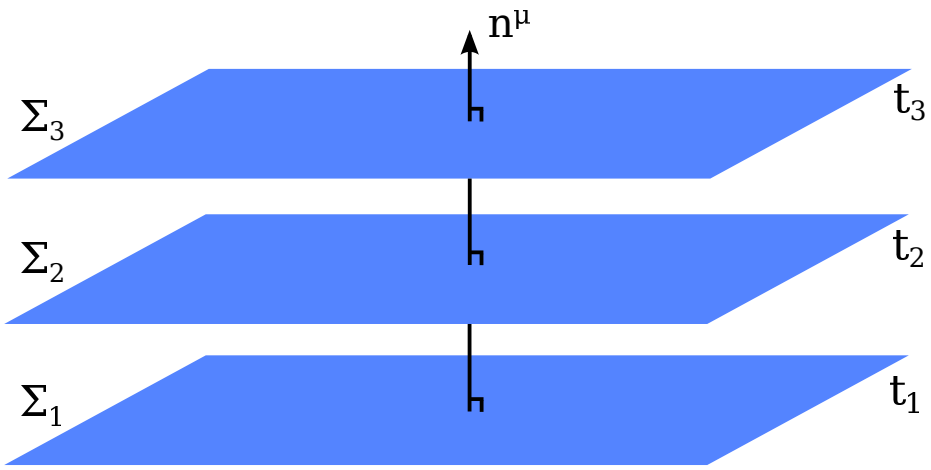



Figure 2.1: Simple foliation of spacetime: the hypersurfaces $\Sigma_1, \Sigma_2, \Sigma_3$ correspond to surfaces of constant time coordinate $t_1, t_2, t_3$ respectively. The four vector, $n^\mu$ is the unit vector normal to the surfaces

$$n_\mu = -\alpha \nabla_\mu t, \tag{2.2}$$

which is the one-form corresponding to the unit vector orthogonal to the hypersurface (See Figure 2.1). Using Equation 2.2 we can now define the projection operator onto the surface of constant $t$, $\Sigma_t$, as

$$\perp^\mu{}_\nu \equiv \delta^\mu{}_\nu + n^\mu n_\nu, \tag{2.3}$$

where $\delta^\mu{}_\nu$ is the Kronecker delta, and $n^\mu = g^{\mu\nu} n_\nu$ is the unit vector normal to the hypersurface corresponding to $n_\nu$. Now, for an arbitrary tangent vector $a^\mu$, the quantity $\perp^\nu{}_\mu a^\mu$ is the projection of that vector onto a surface with constant $t$, $\Sigma_t$.

By applying Equation 2.3 twice to the four-metric $g_{\mu\nu}$, we arrive at the spatial metric induced on the hypersurface, $\Sigma_t$,

$$\gamma_{\mu\nu} = \perp^{\mu'}{}_\mu \perp^{\nu'}{}_\nu g_{\mu'\nu'} = g_{\mu\nu} + n_\mu n_\nu. \tag{2.4}$$

The spatial elements ($\mu\nu = ij$) of Equation 2.4 will form our spatial metric $\gamma_{ij}$ which is used to measure distances on the hypersurface,

$$ds^2|_{\Sigma_t} = \gamma_{ij} dx^i dx^j. \tag{2.5}$$

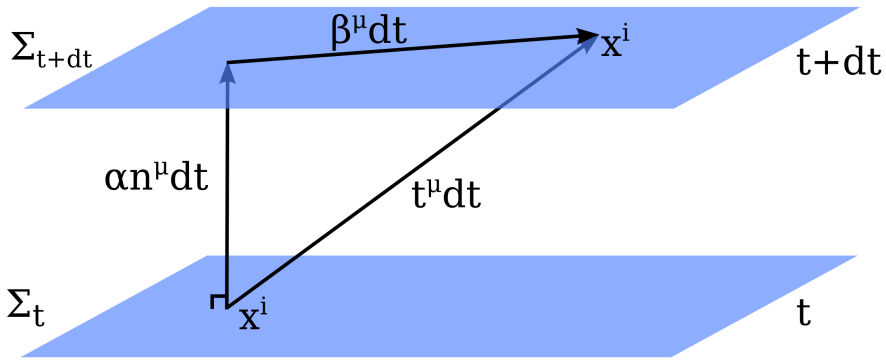



Figure 2.2: 3+1 decomposition of spacetime: between two hypersurfaces, $\Sigma_t$ and $\Sigma_{t+dt}$, the coordinate position $x^i$ is related by the time-like vector $t^\mu$. $t^\mu$ is decomposed into a component tangent to the surface, $\alpha n^\mu$, as well as a purely spatial component component, $\beta^i$. In the language of the ADM formalism, $\alpha$ and $\beta^i$ represent the lapse scalar and shift vector, respectively.

The spatial metric and its inverse, $\gamma^{ij}$, also function as a means to raise and lower indices and calculate inner products of spatial vectors with components wholly on on $\Sigma_t$.

We now look at the time-like vector tangent to the curve of constant coordinate $x^i$ (Figure 2.2) and write it as,

$$t^\mu = \alpha n^\mu + \beta^\mu, \tag{2.6}$$

where we have introduced the purely spatial "shift" four-vector $\beta^\mu = \perp^\mu{}_\nu t^\nu = (0, \beta^i)$. The shift will most commonly appear as simply the three-vector $\beta^i$. With the introduction of these variables, we may now rewrite the spacetime's line element as

$$ds^2 = g_{\mu\nu} dx^\mu dx^\nu = -\alpha^2 dt^2 + \gamma_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt). \tag{2.7}$$

Reading off the terms of Equation 2.7, it is clear that we have the following form for the full metric in the 3+1 formalism,

$$g_{\mu\nu} = \begin{bmatrix} -\alpha^2 + \beta_k \beta^k & \beta_j \\ \beta_i & \gamma_{ij} \end{bmatrix} \tag{2.8}$$

while its inverse is,

$$g^{\mu\nu} = \begin{bmatrix} -1/\alpha^2 & \beta^j/\alpha^2 \\ \beta^i/\alpha^2 & \gamma^{ij} - \beta^i\beta^j/\alpha^2 \end{bmatrix} \tag{2.9}$$

where the inverse of the spatial metric is calculated to satisfy $\gamma^{ik}\gamma_{kj} = \delta^i_j$.

The determinant of the spatial metric $\gamma$ is related to the determinant of the four-metric, $g$ as follows:

$$\sqrt{-g} = \alpha\sqrt{\gamma} \tag{2.10}$$

### 2.1.1 Eulerian vs. Coordinate Observers

With our spacetime foliated, we will now define two types of observer. The first is a coordinate observer that remains stationary with respect to the coordinate system, i.e. its worldline obeys $x^i = \text{const}$; the basis for such an observer is $\{\partial_t, \partial_i\}$. The second observer of interest remains at rest on the spatial slices $\Sigma_t$ and is referred to as an Eulerian observer. The basis for such an observer is given by $\{\mathbf{n}, \partial_i\}$. Put differently, the Eulerian observer at a point $x^i$ on $\Sigma_t$ has a four-velocity in the normal direction, $n^\mu$. As a matter of convention, we will use a bar to refer to the coordinates of Eulerian observers (i.e. $\bar{x}^\mu = (\bar{t}, \bar{x}^i)$) while using $x^\mu = (t, x^i)$ in reference to coordinate observers.

We may transform between these coordinate systems using the following rules:

$$\mathrm{d}\bar{t} = \alpha \mathrm{d}t \tag{2.11}$$

$$\mathrm{d}\bar{x}^i = \mathrm{d}x^i + \beta^i \mathrm{d}t \tag{2.12}$$

In order to transform tensors between these two coordinate systems, we must also define the following Jacobi matrices:

$$\frac{\partial \bar{x}^\mu}{\partial x^\nu} = \begin{bmatrix} \alpha & 0 \\ \beta^i & \delta^i{}_j \end{bmatrix}, \tag{2.13}$$

$$\frac{\partial x^\mu}{\partial \bar{x}^\nu} = \begin{bmatrix} 1/\alpha & 0 \\ -\beta^i/\alpha & \delta^i{}_j \end{bmatrix}. \tag{2.14}$$

By applying the Jacobi matrix (Equation 2.14) to the metric $g_{\mu\nu}$, we can see that the transformed metric has the expected form for an Eulerian reference frame:

$$\bar{g}_{\mu\nu} = \begin{bmatrix} -1 & 0 \\ 0 & \bar{\gamma}_{ij} \end{bmatrix}. \tag{2.15}$$

Since the Eulerian observer is following a worldline tangent to $n^\mu$, we require the following forms for $\bar{n}^\mu$ and $\bar{n}_\mu$

$$\bar{n}^\mu = (1, 0, 0, 0), \tag{2.16}$$

$$\bar{n}_\mu = \bar{g}_{\mu\nu} n^\nu = (-1, 0, 0, 0). \tag{2.17}$$

Now, using the Jacobi matrix (Equation 2.13) to transform the tangent vector, $\bar{n}^\mu$, into the coordinate frame and using the fact that $n_\mu n^\mu = -1$, we arrive at the following forms in the frame of a coordinate observer,

$$n_\mu = (-\alpha, 0, 0, 0), \tag{2.18}$$

$$n^\mu = (1/\alpha, -\beta^i/\alpha). \tag{2.19}$$

With this, we have the basis upon which we will derive the equations of hydrodynamics within general relativity.

# CHAPTER III

# GENERAL RELATIVISTIC HYDRODYNAMICS

In this chapter, we derive the basic form of the equations of general relativistic hydrodynamics within the 3+1 formalism. These equations will be formulated in a manner suitable for a Lagrangian treatment of fluid, in which our numerical variables represent the fluid variables per unit mass. We will then test the limits of our equations: first, in the absence of fluid pressure (and thus hydrodynamical forces) in which they should reduce to the equations of geodesic motion, and then in both the special relativistic and Newtonian limits.

Before we begin, it is important that we relate the motion of fluid in reference frames of coordinate observers, Eulerian observers, and the observers comoving with fluid itself, Lagrangian observers. We begin with a fluid moving with a four-velocity **U**. This four velocity will have the following components in the Eulerian reference frame:

$$\bar{U}^{\mu} = (W, WV^{i}), \tag{3.1}$$

$$\bar{U}_{\mu} = (-W, WV_{i}), \tag{3.2}$$

where we have defined $W \equiv -n_{\mu}U^{\mu} = \alpha U^{0} = -\bar{n}_{\mu}\bar{U}^{\mu} = \bar{U}^{0}$ and $V^{i} = d\bar{x}^{i}/d\bar{t}$. As used here, it is apparent that $W$ is the Lorentz factor between the particle and observer, and $\bar{V}^{i}$ is their relative velocity. Using the normalization of the four-velocity ($\bar{U}^{\mu}\bar{U}_{\mu} = -1$), we arrive at the following equation for the Lorentz factor,

$$W = \frac{1}{\sqrt{1 - \gamma_{ij}V^{i}V^{j}}}. \tag{3.3}$$

By then applying the Jacobi matrices (Equations 2.14 and 2.13) to Equations 3.1 and 3.2, we find that **U** in the coordinate frame takes the following form:

$$U^\mu = \left(\frac{W}{\alpha}, W\left(V^i - \frac{\beta^i}{\alpha}\right)\right) \tag{3.4}$$

$$= \left(\frac{W}{\alpha}, \frac{Wv^i}{\alpha}\right) \tag{3.5}$$

$$U_\mu = (-W(\alpha - \gamma^{ij}\beta_i V_j), WV_i) \tag{3.6}$$

with $v^i = dx^i/dt$ being the components of the coordinate three-velocity. We can now relate the velocities of the coordinate and Eulerian observers as follows,

$$v^i = \alpha V^i - \beta^i. \tag{3.7}$$

We will now consider the case of a Lagrangian observer, i.e. one with a coordinate four velocity $U^\mu$ that is comoving with the fluid. It is in the frame of this observer that we will define the physical properties of our fluid, such as the rest mass, $\rho_0$, or the specific internal energy, $u$. The Lagrangian (or total) time derivative, as measured by this observer, is then:

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\alpha}{W}U^\mu\partial_\mu = \partial_t + v^i\partial_i. \tag{3.8}$$

We will use this total Lagrangian time derivative to describe the change in time of the properties of our fluid parcels as they move through the spacetime.

## *3.1 General Relativistic Hydrodynamics*

We will now provide the basic description of a fluid in general relativity [8, 26]. To begin, we assume a "perfect" fluid: one that does not have dissipation, heat flux, or shear stresses. The stress-energy tensor for such a fluid takes the following form,

$$T_{\mu\nu} = (\rho + P)U_\mu U_\nu + P g_{\mu\nu}, \tag{3.9}$$

where $\rho$ is the energy density in the fluid rest frame, $P$ is the pressure in the fluid rest frame, $U$ is the fluid four-velocity, and $g_{\mu\nu}$ is the four-metric. The energy density $\rho$ can be rewritten as

$$\rho = \rho_0 + u\rho_0 = \rho_0(1 + u), \tag{3.10}$$

where we have broken the energy density into the contribution that results from the rest mass energy density, $\rho_0$, and the contribution from the fluid's specific internal energy, $u$. It will also prove convenient to define the fluid's relativistic specific enthalpy, $\omega$, as

$$\omega = 1 + u + \frac{P}{\rho_0}. \tag{3.11}$$

Having introduced Equations 3.10 and 3.11 we can now recast the stress-energy tensor into a more convenient form:

$$T_{\mu\nu} = \rho_0 \omega U_\mu U_\nu + P g_{\mu\nu}. \tag{3.12}$$

We will now assume that our fluid consists of only one component with number density, $N(x^\mu)$, and particle mass, $m_B$. This will allow us to express the rest-mass density as

$$\rho_0 = m_B N, \tag{3.13}$$

and then obtain the following form for the the fluid's four-current,

$$J^\mu = \rho_0 U^\mu. \tag{3.14}$$

With these definitions we can now obtain the equations of motion from the conserved properties of the fluid. First, we will consider the conservation of baryon number,

$$\nabla_\mu J^\mu = \nabla_\mu (\rho_0 U^\mu) = 0, \tag{3.15}$$

which will eventually lead to our continuity equation. Second, we will consider the conservation of the stress energy,

$$\nabla_\mu T^{\mu\nu} = 0, \tag{3.16}$$

to arrive at the relativistic form of Euler's equations for energy and momentum. To this end, Equation 3.16 must be projected along the time-like vector, $t^\mu$ (Equation 2.6), and on to the space-like hypersurface using $\gamma_{ij}$ (Equation 2.4).

### 3.1.1 The Equation of State

We must supplement our system of equations in order to close the system. For this, we will need an Equation of State (EoS) relating the pressure, $P$; the density, $\rho_0$; and the internal energy, $u$. To satisfy our requirements, we will need an equation of the form $f(P, \rho_0, u) = 0$. In this work we make use of two relatively simple equations of state: that of an ideal fluid and that of a polytropic gas [8].

#### *3.1.1.1 Ideal Gas Equation of State*

It is often useful to adopt the "Γ-law" equation of state, which describes an ideal, relativistic, non-degenerate, adiabatic, monoatomic gas with adiabatic index Γ:

$$P = (\Gamma - 1)\rho_0 u. \tag{3.17}$$

Numerically, Equation 3.17 is relatively simple and can be used to approximate various states of matter. For example, $\Gamma \approx 2$ can model the stiff nuclear matter of a neutron star, $\Gamma = 5/3$ can model a nonrelativistic ideal gas (e.g. a common star), and $\Gamma = 4/3$ can model a highly relativistic gas (e.g. a thermal radiation dominated supermassive star).

#### *3.1.1.2 Polytropic Equation of State*

For a gas undergoing purely isentropic flow, the first law of thermodynamics can be used to rewrite Equation 3.17 into the equation of state governing a polytrope:

$$P = K\rho_0^{\Gamma}, \tag{3.18}$$

with $\Gamma = 1 + 1/n$, $n$ representing the polytropic index, and $K$ being the gas constant determined by the entropy of the gas. This equation will break down in the case of a nonisentropic flow (e.g. in the presence of a shock) as the gas constant, $K$, will no longer be constant throughout the fluid.

### 3.1.2 The Speed of Sound

Another useful quantity to describe our fluid is the speed at which signals propagate through the medium, the sound speed $c_s$. The speed of sound is defined by taking the partial derivative of the pressure with respect to the internal energy density taken at constant entropy $s$,

$$c_s^2 \equiv \left.\frac{\partial P}{\partial \rho}\right|_s . \tag{3.19}$$

By applying Equation 3.19 to the Γ-law equation of state (Equation 3.17) we get the following form,

$$c_s = \sqrt{\frac{\Gamma p}{\rho_0 \omega}}. \tag{3.20}$$

Taking the Newtonian limit, where $\rho_0\omega \to \rho$, Equation 3.20 takes the usual form for an ideal gas, $c_s \to \sqrt{\Gamma P/\rho}$. It is also convenient to introduce the ratio of the fluid's flow velocity to the local speed of sound, the fluid's Mach number $\mathcal{M}$:

$$\mathcal{M} = \frac{||V||}{c_s} = \frac{\gamma_{ij} V^i V^j}{c_s}. \tag{3.21}$$

## *3.2 The Relativistic Hydrodynamic Evolution Equations*

We will now derive the Lagrangian evolution equations for a fluid in the 3+1 formalism [37, 66]. We achieve this by first splitting the local conservation equations for both stress-energy and rest-mass (Equations 3.15 and 3.16, respectively) using the 3+1 formalism and then recasting the time derivatives into the total Lagrangian form (Equation 3.8). For simplicity, we will now introduce the following fluid variables as observed by an Eulerian observer:

$$D = -n_\mu J^\mu = -n_\mu(\rho_0 U^\mu) = \rho_0 W \tag{3.22}$$

$$E = n^\mu n^\nu T_{\mu\nu}/D = \omega W - P/D \tag{3.23}$$

$$S^i = -\gamma^{ij} n^\mu T_{\mu j}/D = \omega W V^i \tag{3.24}$$

$$S^{ij} = \gamma^{\mu i}\gamma^{\nu j} T_{\mu\nu}/D = \omega W V^i V^j + (P/D)\gamma^{ij} \tag{3.25}$$

where $D$ is the mass density, $E$ is the total relativistic specific energy, $S^i$ is the relativistic specific three-momentum, and $S^{ij}$ is the specific spatial stress tensor.

### 3.2.1 Choosing a Suitable Density

We will now derive a suitable density variable from the continuity equation, Equation 3.15. Using the following identity for the divergence of a four-vector $A^\mu$ [38],

$$\nabla_\mu A^\mu = \frac{1}{\sqrt{-g}}\partial_\mu(\sqrt{-g}A^\mu), \tag{3.26}$$

we rewrite Equation 3.15 using Equations 2.10, 3.22, and 3.6 as

$$\partial_\mu(\alpha\sqrt{\gamma}\rho_0 U^\mu) = \partial_t(\alpha\sqrt{\gamma}\rho_0 U^0) + \partial_i(\alpha\sqrt{\gamma}\rho_0 U^0 v^i) = 0 \tag{3.27}$$

$$= \partial_t(\sqrt{\gamma}D) + \partial_i(\sqrt{\gamma}Dv^i) = 0 \tag{3.28}$$

Now, using the total Lagrangian time derivative (Equation 3.8) and Equation 3.22, we rewrite Equation 3.28 as

$$\frac{\mathrm{d}D}{\mathrm{d}t} + D\partial_i v^i + D\frac{\mathrm{d}}{\mathrm{d}t}\ln\sqrt{\gamma} = 0. \tag{3.29}$$

This equation contains a term resulting from the total mass flux of the volume element $\mathrm{d}V$ as well as a second term due to the time variation of the volume $\mathrm{d}V$. The implicit nature of of the term $\mathrm{d}/\mathrm{d}t(\ln\sqrt{\gamma})$ will have negative effects on our numerical scheme [37, 66], so we seek a form more like that of a traditional Newtonian continuity equation for SPH. For this, we define a new density variable that accounts for the spacetime geometry,

$$D^\star = \sqrt{\gamma}D. \tag{3.30}$$

Then by substituting it into Equation 3.28, and using Equation 3.8 to convert to the total Lagrangian time derivative, we arrive at the following continuity equation

$$\frac{\mathrm{d}D^\star}{\mathrm{d}t} + D^\star \partial_i v^i = 0. \tag{3.31}$$

#### *3.2.1.1 Conservation of Rest Mass*

We will now show that there exists an appropriately conserved mass variable related to our density $D^\star$ [8]. This mass will eventually be used to weight our discretization procedure. We begin by integrating our continuity equation, Equation 3.15, over a region of spacetime, $\Omega$,

$$\int_\Omega \nabla_\mu(\rho_0 U^\mu)\sqrt{-g}\mathrm{d}^4x = 0 \tag{3.32}$$

By applying Gauss's theorem, we can replace the integrated divergence in the above equation with an integral over the boundary, $\delta\Omega$,

$$\int_{\delta\Omega} \rho_0 U^\mu \mathrm{d}^3\Sigma_\mu = 0, \tag{3.33}$$

where $\Sigma_\mu = \epsilon\mathcal{N}_\mu\sqrt{\gamma}\mathrm{d}^3x$. We have introduced the unit normal vector pointing outward from our surface, $\mathcal{N}_\mu$, as well has the the factor $\epsilon$ where $\epsilon = 1$ or $\epsilon = -1$ for a timelike surface or spacelike surface $\delta\Omega$, respectively.

We may now define our Gaussian surface to be bounded by two spacelike hypersurfaces, $\Sigma_1$ and $\Sigma_2$, as well as a timelike hypersurface that resides entirely outside of our matter source. Because it lies in the vacuum outside of our matter source, this timelike surface will not contribute to our surface integral. On the surface $\Sigma_2$, the unit vector normal to the surface of integration, $\mathcal{N}_\mu$, corresponds to the future pointing normal vector to our spacetime foliation, $n_\mu$, while on the surface $\Sigma_1$ it will correspond to $-n_\mu$ as it the the normal vector pointing into the past. Since $n_\mu U^\mu = -\alpha U^0 = -W$, we may now rewrite our surface integral as

$$\int_{\Sigma_1} \sqrt{\gamma} W \rho_0 \mathrm{d}V - \int_{\Sigma_2} \sqrt{\gamma} W \rho_0 \mathrm{d}V = 0 \tag{3.34}$$

where $\mathrm{d}V$ is a differential volume element defined on our two spacelike surfaces. This now implies the conservation of rest mass defined as

$$m = \int_\Sigma \sqrt{\gamma} W \rho_0 \mathrm{d}V = \int_\Sigma D^\star \mathrm{d}V. \tag{3.35}$$

### 3.2.2 The Momentum Equation

To derive our equations for the evolution of the energy and momentum we will begin by applying the following identity for the four-divergence of a symmetric tensor

$$\nabla_\mu A^\mu_\nu = \frac{1}{\sqrt{-g}} \partial_\mu (\sqrt{-g} A^\mu_\nu) - \frac{1}{2} \partial_\nu g_{\mu\sigma} A^{\mu\sigma} = 0 \tag{3.36}$$

to conservation of energy-momentum (Equation 3.16), yielding

$$\partial_\mu (\alpha \sqrt{\gamma} T^\mu_\nu) - \frac{\alpha \sqrt{\gamma}}{2} \partial_\nu g_{\mu\sigma} T^{\mu\sigma} = 0. \tag{3.37}$$

At this point, we will also need the stress energy tensor (Equation 3.12) with both mixed and upper indices:

$$T^\mu_\nu = g^{\mu\sigma} T_{\sigma\nu} = \rho_0 \omega U^\mu U_\nu + P \delta^{\mu\nu} \tag{3.38}$$

$$T^{\mu\nu} = g^{\mu\sigma} g^{\nu\lambda} T_{\sigma\lambda} = \rho_0 \omega U^\mu U^\nu + P g^{\mu\nu} \tag{3.39}$$

To arrive at the equations for momentum evolution, we will begin by taking the relevant spatial components ($\nu = i$) of Equation 3.37,

$$\partial_\mu (\alpha \sqrt{\gamma} T^\mu_i) = \frac{\alpha \sqrt{\gamma}}{2} \partial_i g_{\mu\sigma} T^{\mu\sigma}. \tag{3.40}$$

To rewrite the the left hand side, we will make use of the following terms from Equation 3.38 (substituting in Equations 3.5 and 3.6):

$$T^0_i = \frac{\rho_0 \omega W^2 V_i}{\alpha} = \frac{D S_i}{\alpha} \tag{3.41}$$

$$T^j_i = \frac{\rho_0 \omega W^2 V_i v^j}{\alpha} + P \delta^j_i = \frac{D S_i v^j}{\alpha} + P \delta^j_i \tag{3.42}$$

Using these terms of the stress energy tensor along with Equation 3.30, we obtain:

$$\partial_t (D^\star S_i) + \partial_j (D^\star S_i v^j) + \partial_j (\alpha \sqrt{\gamma} P \delta^j_i) = \frac{\alpha \sqrt{\gamma}}{2} \partial_i g_{\mu\sigma} T^{\mu\sigma}. \tag{3.43}$$

We must now recast the partial time derivative, $\partial_t$, into the Lagrangian form. Using Equation 3.8 and our continuity equation (Equation 3.31), our momentum equation is now of the form:

$$\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}P) = \frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_i g_{\mu\sigma}T^{\mu\sigma}. \tag{3.44}$$

### 3.2.3 The Energy Equation

In order to find the evolution equation for our energy variable, we will next turn our attention to the time component of Equation 3.37 ($\nu = 0$),

$$\partial_\mu(\alpha\sqrt{\gamma}T^\mu_0) = \frac{\alpha\sqrt{\gamma}}{2}\partial_0 g_{\mu\sigma}T^{\mu\sigma} \tag{3.45}$$

Clearly, we will need the following elements of the mixed index stress energy tensor (Equation 3.38):

$$\begin{aligned} T^0_0 &= -\frac{\rho_0\omega W^2}{\alpha}(\alpha - \beta^i V_i) + P \\ &= -\frac{D(\alpha E - \beta^i S_i)}{\alpha} \\ &= -\frac{D\bar{E}}{\alpha} \end{aligned} \tag{3.46}$$

$$\begin{aligned} T^i_0 &= -\frac{\rho_0\omega W^2 v^i}{\alpha}(\alpha - V_j\beta^j)v^i \\ &= -\left[\frac{D(\alpha E - \beta^i S_i)}{\alpha} + P\right]v^i \\ &= -\left[\frac{D\bar{E}}{\alpha} + P\right]v^i \end{aligned} \tag{3.47}$$

where the variable

$$\bar{E} = \alpha E - \beta^i S_i \tag{3.48}$$

has been introduced for convenience [66]. These terms allow us to rewrite the left hand side of Equation 3.45 to obtain:

$$\partial_t(D^\star\bar{E}) + \partial_i(D^\star\bar{E}v^i) + \partial_i(\alpha\sqrt{\gamma}Pv^i) = -\frac{\alpha\sqrt{\gamma}}{2}\partial_t g_{\mu\sigma}T^{\mu\sigma}. \tag{3.49}$$

In the same manner as the spatial case, we now use Equation 3.8 and our continuity equation (3.31) to rewrite our energy evolution equation:

$$\frac{\mathrm{d}\bar{E}}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}Pv^i) = -\frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_t g_{\mu\sigma}T^{\mu\sigma}. \tag{3.50}$$

### 3.2.4 Gravitational Terms

Equations 3.31, 3.44, and 3.50 now form a complete set of Lagrangian evolution equations for a relativistic fluid:

$$\begin{aligned}
\frac{\mathrm{d}D^\star}{\mathrm{d}t} + D^\star\partial_i v^i &= 0 && (3.51)\\
\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}P) &= \frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_i g_{\mu\sigma}T^{\mu\sigma} && (3.52)\\
\frac{\mathrm{d}\bar{E}}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}Pv^i) &= -\frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_t g_{\mu\sigma}T^{\mu\sigma} && (3.53)
\end{aligned}$$

All that remains is the 3+1 decomposition of the right hand side terms. These terms represent the action that the spacetime itself will have on the fluid. The terms appearing in Equations 3.44 and 3.50 are both of the form,

$$\pm\frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_\lambda g_{\mu\sigma}T^{\mu\sigma}, \tag{3.54}$$

for $\lambda = i, t$, respectively. To cast this into the 3+1 variables, we will need to calculate the various components of $T^{\mu\nu}$ that appear:

$$\begin{aligned}
T^{00} &= \frac{\rho_0\omega W^2 - P}{\alpha^2} = \frac{DE}{\alpha^2} && (3.55)\\
T^{0i} &= \frac{\rho_0\omega W^2}{\alpha^2}(\alpha V^i - \beta^i) + \frac{P\beta^i}{\alpha^2} = \frac{DS^i}{\alpha} - \frac{DE\beta^i}{\alpha^2} && (3.56)\\
T^{ij} &= \frac{\rho_0\omega W^2}{\alpha^2}(\alpha V^i - \beta^i)(\alpha V^j - \beta^j) + P\gamma^{ij} - \frac{P\beta^i\beta^j}{\alpha^2}\\
&= \frac{DE\beta^i\beta^j}{\alpha^2} - \frac{D(S^i\beta^j + S^j\beta^i)}{\alpha} + DS^{ij} && (3.57)
\end{aligned}$$

Equation 3.54 can now be rewritten as,

$$\pm\frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_\lambda g_{\mu\sigma}T^{\mu\sigma} = \mp E\partial_\lambda\alpha \pm S_j\partial_\lambda\beta^j \pm \frac{\alpha}{2}S^{jk}\partial_\lambda\gamma_{jk} \tag{3.58}$$

With this we arrive at the final form of our Lagrangian fluid equations in the 3+1 decomposition:

$$\frac{\mathrm{d}D^\star}{\mathrm{d}t} + D^\star \partial_i v^i = 0 \tag{3.59}$$

$$\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}P) = -E\partial_i\alpha + S_j\partial_i\beta^j + \frac{\alpha}{2}S^{jk}\partial_i\gamma_{jk} \tag{3.60}$$

$$\frac{\mathrm{d}\bar{E}}{\mathrm{d}t} + \frac{1}{D^\star}\partial_i(\alpha\sqrt{\gamma}Pv^i) = E\partial_t\alpha - S_j\partial_t\beta^j - \frac{\alpha}{2}S^{jk}\partial_t\gamma_{jk} \tag{3.61}$$

Together with an equation of state (Subsection 3.1.1), we now have a complete set of equations to describe a relativistic fluid in a spacetime decomposed via the 3+1 formalism.

### 3.2.5 Comparison with Previous Formalisms

We will now compare our evolution equations with those of previous formalisms of relativistic hydrodynamics. Perhaps the most commonly used [48, 52] form of the general relativistic hydrodynamics equations is that of the Valencia group [43, 7]. This formulation is written in a conservative form which is well suited for Eulerian methods. These evolution equations as a flux conservative, hyperbolic system take the form:

$$\frac{1}{\sqrt{-g}}\left(\partial_t(\sqrt{\gamma}\vec{U}) + \partial_i(\sqrt{-g}\vec{F}^i)\right) = \vec{S} \tag{3.62}$$

where the conserved variables represented by the vector $\vec{U}$ take the following form:

$$\tilde{D} = \rho_0 W \tag{3.63}$$

$$\tilde{S}_i = \rho_0 \omega W^2 V_i \tag{3.64}$$

$$\tilde{\tau} = D(E-1) = \rho_0\omega W^2 - P - D \tag{3.65}$$

and the flux terms, $\vec{F}^i$, and source terms, $\vec{S}$, take the form:

$$\vec{F}^i = \left(\frac{Dv^i}{\alpha}, \frac{S_j v^i}{\alpha} + P\delta^i_j, \frac{\tau v^i}{\alpha} + PV^i\right) \tag{3.66}$$

$$\vec{S} = \left(0, T^{\mu\nu}\partial_\mu g_{\nu j} - \Gamma^\delta_{\nu\mu} g_{\delta j}, \alpha\left(T^{\mu t}\partial_\mu \ln\alpha - T^{\mu\nu}\Gamma^t_{\nu\mu}\right)\right) \tag{3.67}$$

One can easily see the differences present between our formula and that of the Valencia group. In particular, their conservative variables in terms of the conservative fluid variables we have outlined in this chapter ($D^\star, S_i, \bar{E}$) are:

$$\tilde{D} = \frac{D^\star}{\sqrt{\gamma}} \tag{3.68}$$

$$\tilde{S}_i = DS_i \tag{3.69}$$

$$\tilde{\tau} = D(E-1) = D(\frac{\bar{E} + \beta^i S_i}{\alpha} - 1) \tag{3.70}$$

These differences largely arise from the Lagrangian nature of our equations, where we have taken our conservative variables to be defined "per unit D" as well as our particular choice of a density variable weighted by the determinant of the spatial metric.

### 3.2.6 Geodesic Limit

In the limit of vanishing hydrodynamic forces, the equations of hydrodynamics should reduce to that of geodesic motion, i.e. the motion of non-interacting particles moving through the spacetime. To test our hydrodynamics equations in this limit, we begin by choosing an EoS with the property that $P = u = 0$, thereby nullifying the hydrodynamical interactions. This will also transform the following variables:

$$\omega \rightarrow 1 \tag{3.71}$$

$$S_i \rightarrow WV_i = U_i \tag{3.72}$$

$$\bar{E} \rightarrow W(\alpha - V_i\beta^i) = -U_0 \tag{3.73}$$

$$T^{\mu\nu} \rightarrow \rho_0 U^\mu U^\nu \tag{3.74}$$

As a result the evolution equations (5.16 and 5.18) now take the form:

$$\frac{\mathrm{d}U_i}{\mathrm{d}t} = \frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_i g_{\mu\nu} T^{\mu\nu} \tag{3.75}$$

$$\frac{\mathrm{d}(-U_0)}{\mathrm{d}t} = -\frac{\alpha\sqrt{\gamma}}{2D^\star}\partial_t g_{\mu\nu} T^{\mu\nu} \tag{3.76}$$

These can be combined into the single equation

$$\frac{\mathrm{d}U_\lambda}{\mathrm{d}t} = \frac{\alpha\sqrt{\gamma}\rho_0}{2D^\star}\partial_\lambda g_{\mu\nu}U^\mu U^\nu = \frac{\alpha}{2W}\partial_\lambda g_{\mu\nu}U^\mu U^\nu. \tag{3.77}$$

Now, we use the fact that $\mathrm{d}t = (\mathrm{d}t/\mathrm{d}\tau)d\tau = (W/\alpha)\mathrm{d}\tau$ to reach the form,

$$\frac{\mathrm{d}U_\lambda}{\mathrm{d}\tau} = \frac{1}{2}\partial_\lambda g_{\mu\nu}U^\mu U^\nu. \tag{3.78}$$

Rewriting $U_\lambda = g_{\lambda\mu}U^\mu$, it can be seen that:

$$\frac{\mathrm{d}U_\lambda}{\mathrm{d}\tau} = \frac{\mathrm{d}(g_{\lambda\mu}U^\mu)}{\mathrm{d}\tau} = g_{\lambda\mu}\frac{\mathrm{d}U^\mu}{\mathrm{d}\tau} + U^\mu\frac{\mathrm{d}g_{\lambda\mu}}{\mathrm{d}\tau}. \tag{3.79}$$

By applying the identity $\mathrm{d}/\mathrm{d}\tau = U^\mu\partial_\mu$ to the second term, we see that:

$$U^\mu\frac{\mathrm{d}g_{\lambda\mu}}{\mathrm{d}\tau} = U^\mu U^\nu\partial_\nu g_{\lambda\mu} = \frac{1}{2}\left(\partial_\nu g_{\lambda\mu} + \partial_\mu g_{\lambda\nu}\right)U^\mu U^\nu. \tag{3.80}$$

We now use Equations 3.79 and 3.80 to rewrite Equation 3.78 as,

$$g_{\lambda\mu}\frac{\mathrm{d}U^\mu}{\mathrm{d}\tau} = -\frac{1}{2}\left(\partial_\nu g_{\lambda\mu} + \partial_\mu g_{\lambda\nu} - \partial_\lambda g_{\mu\nu}\right)U^\mu U^\nu. \tag{3.81}$$

Now, by simply multiplying both sides by the metric inverse $g^{\sigma\lambda}$, we recover the equation of geodesic motion [16]:

$$\frac{\mathrm{d}U^\mu}{\mathrm{d}\tau} = -\frac{g^{\sigma\lambda}}{2}\left(\partial_\nu g_{\lambda\mu} + \partial_\mu g_{\lambda\nu} - \partial_\lambda g_{\mu\nu}\right)U^\mu U^\nu \tag{3.82}$$

$$= -\Gamma^\sigma_{\mu\nu}U^\mu U^\nu \tag{3.83}$$

### 3.2.7 Special Relativistic Limit

In this section we take the special relativistic limit of our equations and compare them to previous formulations [60, 47]. To achieve this we can simply set the spacetime to that of the Minkowski spacetime, $g_{\mu\nu} = \eta_{\mu\nu} = diag(-1, 1, 1, 1)$. In this case, the

ADM variables are $\alpha = 1, \beta^i = 0, \gamma_{ij} = \delta_{ij}$ and our hydrodynamic quantities become:

$$D^\star \rightarrow D = \rho_0 W \tag{3.84}$$

$$V^i \rightarrow v^i \tag{3.85}$$

$$W \rightarrow \frac{1}{\sqrt{1 - \delta_{ij} v^i v^j}} \tag{3.86}$$

$$S^i \rightarrow \omega W v^i \tag{3.87}$$

$$\bar{E} \rightarrow E = \omega W - \frac{P}{D} \tag{3.88}$$

As a result the evolution equations take the form

$$\frac{\mathrm{d}D}{\mathrm{d}t} + D\partial_i v^i = 0 \tag{3.89}$$

$$\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{1}{D}\partial_i(P) = 0 \tag{3.90}$$

$$\frac{\mathrm{d}E}{\mathrm{d}t} + \frac{1}{D}\partial_i(Pv^i) = 0 \tag{3.91}$$

which have the same form as previous special relativistic formulations [60, 47].

### 3.2.8 Newtonian Limit

We will now check the validity of our evolution equations in the Newtonian limit. Starting from the special relativistic equations (Equations 3.89, 3.90, and 3.91), we further simplify our system by noting that the following variables will change in the limit that $||v||^2 << 1$, $u << 1$, and $P << \rho$:

$$W \rightarrow 1 + v^2/2 \tag{3.92}$$

$$D \rightarrow \rho \tag{3.93}$$

$$S^i \rightarrow v^i \tag{3.94}$$

$$E \rightarrow u + v^2/2 \tag{3.95}$$

In this limit we are left with the follow evolution equations,

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} + \rho\partial_i v^i = 0 \tag{3.96}$$

$$\frac{\mathrm{d}v^i}{\mathrm{d}t} + \frac{1}{\rho}\partial_i P = 0 \tag{3.97}$$

$$\frac{\mathrm{d}E}{\mathrm{d}t} + \frac{1}{\rho}\partial_i (P v^i) = 0 \tag{3.98}$$

Alternatively, we can perform a change of variable on Equation 3.98, to obtain

$$\frac{\mathrm{d}u}{\mathrm{d}t} + \frac{P}{\rho}\partial_i v^i = 0. \tag{3.99}$$

With Equations 3.96, 3.97, and 3.99 we see that, in this limit, our system corresponds to the standard prescription for Newtonian SPH [69].

## *3.3 Conservative Variables to Primitive Variables*

To close our system, we must use a relationship between the pressure, $P$, the internal energy, $u$, and the density, $\rho_0$ – the equation of state (Chapter 3.1.1). With this relationship, we will be able to recover the primitive variables that appear on the right hand side of our equations $(W, \rho_0, v^i, P)$ from the evolved variables $(D^\star, S_i, \bar{E})$. In order to do this, we will need to invert the following system of equations (Equations 3.30, 3.24, 3.48, and 3.3):

$$D^\star = \sqrt{\gamma} W \rho_0 \tag{3.100}$$

$$S_i = \frac{W\omega}{\alpha}(v_i + \beta_i) = W\omega V_i \tag{3.101}$$

$$\bar{E} = \alpha E - \beta^i S_i = \alpha(\omega W - P/D) - \beta^i S_i \tag{3.102}$$

$$W = \frac{1}{\sqrt{1 - V_i V^i}} \tag{3.103}$$

In order to do this, we will use the following root finding procedure. We will first use the equation of state to write the enthalpy as a function of $\rho_0$ and $P$,

$$\omega = \omega(\rho_0, P). \tag{3.104}$$

We will also write our primitive variables as a function of the enthalpy:

$$W(\omega) = \sqrt{1 + S_i S^i / \omega^2} \tag{3.105}$$

$$\rho_0(\omega) = \frac{D^\star}{\sqrt{\gamma} W(\omega)} \tag{3.106}$$

$$v^i(\omega) = \frac{\alpha S^i}{\omega W(\omega)} - \beta^i \tag{3.107}$$

$$P(\omega) = \frac{D^\star}{\alpha \sqrt{\gamma}} \left( \omega W(\omega) - \bar{E} - S_i \beta^i \right). \tag{3.108}$$

In order to invert this system, we will choose a guess for the value of the enthalpy, $\omega'$. This is normally taken to be the value of the enthalpy from the previous timestep. With this we are free to calculate the values of $P' = P(\omega')$ and $\rho_0' = \rho_0(\omega')$ using Equations 3.108 and 3.106. We then calculate the updated value of the enthalpy, $\omega(\rho_0', P')$. With this, we will check to see if our convergence criteria is satisfied, i.e.

$$\omega(\rho_0', P') - \omega' < \epsilon \, \min(\omega', \omega), \tag{3.109}$$

where $\epsilon$ is the relative error allowed in our root finding method, normally taken to be $10^{-5}$. If this criteria is not satisfied, we will update our guess to a new value and repeat the root finding procedure. In practice, we use Steffensen's Method[1] to choose our updated value of $\omega'$.

We find that we are normally able to solve for the correct value of $\omega$ with only a couple iterations of our root finding method due to the fact that the value of $\omega$ only varies mildly with time.

[1]Steffensen's Method is simply an accelerated version of the traditional Newton's Method which will converge faster under normal conditions.

# CHAPTER IV

# SMOOTHED PARTICLE HYDRODYNAMICS

Throughout the vast set of astrophysical processes important physics can span a large range of length and time scales. Many such astrophysical scenarios are the very definition of what computational physicists refer to as "multi-scale" and "multi-physics." Typically, when these hydrodynamical processes operate around compact objects they exhibit the following noteworthy properties [59]:

- Fluid boundaries are generally non-existent. As a result of the interactions between gravitational and fluid forces, it is common for matter to become both locally concentrated and exceptionally diffuse. As a result, simulations of these systems require a high level of spatial resolution and adaptivity. Traditional Eulerian schemes often employ adaptive mesh refinement (AMR) to solve this problem (e.g the Berger-Oliger AMR [10] method used in Carpet [63]).

- Shocks often play a critical role in the long term dynamical evolution of the fluid in the vicinity of a compact object. Physically, a shock is a thin transition between a region of supersonic flow and a region of subsonic flow. The propagation of such a shock wave may lead to infinite values for the gradient of variables characterizing the fluid, so they must be treated with care in any numerical scheme.

- The values of physical quantities can vary dramatically over the computational domain. Therefore, the robustness of the numerical scheme becomes very important to preserve the accuracy of results. Often this necessitates the use of multiple levels of time refinement to integrate the values of the fluid quantities

to both preserve accuracy in some regions and avoid excessive computation in others.

In choosing a numerical scheme to describe the underlying fluid, one must choose which among these properties are most important to characterizing the system of interest. In this chapter, we will focus on the method known as "Smoothed Particle Hydrodynamics" (SPH) ([42], [28]). Smoothed particle hydrodynamics is a Lagrangian method that solves the equations of hydrodynamics on a completely unstructured mesh. This mesh is advected through the computational domain along with the fluid flow; each "particle" of the mesh representing a parcel of the fluid with fixed mass, $m$. As a result, SPH conserves mass exactly, while also adapting its resolution in proportion to the local fluid density. In this chapter we will develop the SPH interpolation method as it is traditionally performed for flat space. Unlike the method outlined in [37], where a correction factor of $1/\sqrt{\gamma}$ was applied to the SPH interpolation, our SPH interpolation will take place in a locally flat frame [66, 61]. In doing this, we assume that the local radius of curvature of the spacetime is large relative to the length scale over which our particles are smoothed. As a result, we will be able to apply the traditional SPH summations in a curved spacetime.

## *4.1 Kernel Based Interpolation*

In traditional SPH, we advect the mesh each timestep by using the velocity of the local fluid, $\vec{v}$. Since the points of this mesh (the "particles") retain no regular structure, finite differencing is of little use and we must prescribe a way to calculate both functions and their derivatives at the locations of the particles. To this end, SPH methods use a kernel function to perform the interpolation of both fluid variables and their derivatives onto the mesh of particles. This allows us to transform the continuum partial differential equations into a series of ordinary differential equations that are defined locally on the individual particles and are easily numerically integrable.

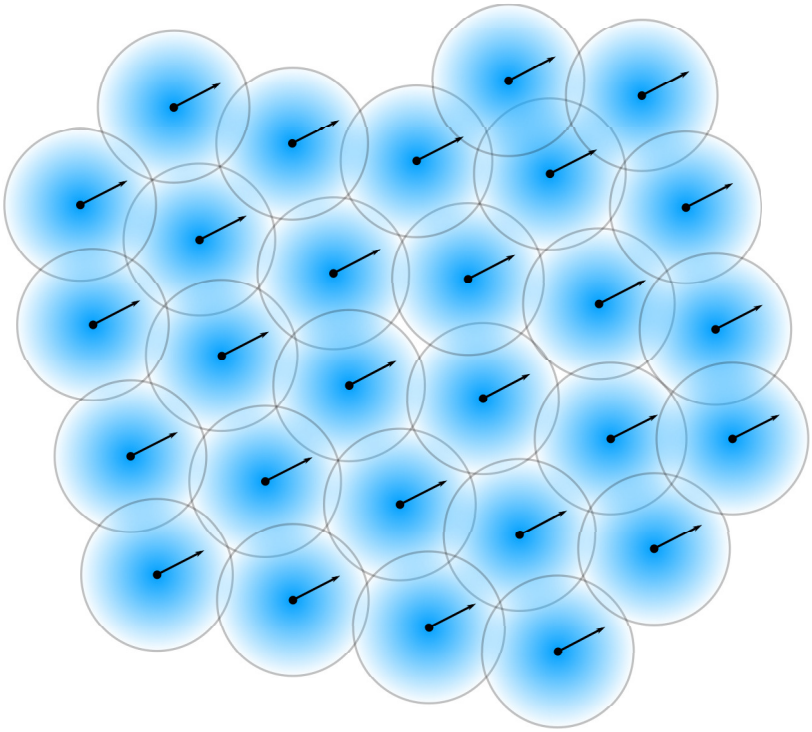


Figure 4.1: Schematic of the SPH method: In this figure, each parcel of fluid is represented by a particle that is being advected through the domain with a velocity represented by the arrows. Each particle has a finite radius of influence as indicated by the circles.

### 4.1.1 Interpolating Scalar Functions

Perhaps most fundamental to the SPH method is the method by which we map a scalar function on to the mesh of particles. We start with the trivial identity describing the scalar function $f(\vec{r})$,

$$f(\vec{r}_a) = \int f(\vec{r}_b)\delta(\vec{r}_a - \vec{r}_b)dV_b, \tag{4.1}$$

where $\delta$ is the Dirac delta function and $\vec{r}_a$ is the position of particle $a$. We will also introduce a normalized kernel function, $W$, with smoothing length, h, such that,

$$\lim_{h\to 0} W(\vec{r}, h) = \delta(\vec{r}). \tag{4.2}$$

By Taylor expansion of $W(\vec{r}_a - \vec{r}_b, h)$, we see that $f(\vec{r}_a)$ will take the following form for symmetric (i.e. $W(\vec{r}_a - \vec{r}_b, h) = W(\vec{r}_b - \vec{r}_a, h)$) kernels

$$f(\vec{r}_a) = \int f(\vec{r}_b) W(|\vec{r}_a - \vec{r}_b|, h) dV_b + \mathcal{O}(h^2). \tag{4.3}$$

The kernel function $W$ will be explained at length in Section 4.1.4. This second order accurate approximation arises from the fact that the gradient of the kernel vanishes at $\vec{r}_a = \vec{r}_b$ for kernel functions symmetric in $\vec{r}_a$ and $\vec{r}_b$.

In order to cast Equation 4.3 into a form suitable for discretization, we must replace the integral with a sum over all grid points. In order to do this, we note that the total mass $m$ within a volume element $\mathrm{d}V$ is given by Equation 3.35 as $m = D^\star dV$. Using this relation, we may estimate the volume element in Equation 4.3 as $\mathrm{d}V \approx \Delta V = m/D^\star$. This substitution will now allow us to replace the integral with a sum over all grid points:

$$\begin{aligned} f_{SPH}(\vec{r}_a) &\approx \sum_b \Delta V_b f_b W(|\vec{r}_a - \vec{r}_b|, h) \\ &\approx \sum_b \frac{m_b}{D_b^\star} f_b W(|\vec{r}_a - \vec{r}_b|, h), \end{aligned} \tag{4.4}$$

where the subscript $f_b$ represents the value at the position of particle $b$ ($f_b = f(\vec{r}_b)$).

By applying Equation 4.4 to our density-like variable $D^\star$, we arrive at the following approximation of the relativistic fluid density,

$$D^\star_{SPH}(\vec{r}) \approx \sum_b m_b W(|\vec{r}_a - \vec{r}_b|, h). \tag{4.5}$$

One can easily see that Equation 4.5 implies the local fluid mass density at any given location is simply the kernel weighted sum of mass contributions from the other particles forming the mesh.

### 4.1.2 Interpolating Spatial Derivatives

Next, we must develop a treatment for the spatial derivatives of the fluid variables at the locations of our particles. We arrive at a suitable form of the gradient by simply

taking the gradient ($\partial_a \equiv \partial/\partial\vec{r}_a$) of Equation 4.4 and noting that only the kernel function, $W(\vec{r}_a - \vec{r}_b, h)$, exhibits a dependence on the position vector, $\vec{r}_a$ [18] . This yields the remarkably simple form,

$$\nabla f_{SPH}(\vec{r_a}) = \sum_b \frac{m_b}{D_b^\star} f_b \partial_a W(|\vec{r_a} - \vec{r_b}|, h), \tag{4.6}$$

which is simply the local values of the function, $f_b$, multiplied by the derivative of our kernel function, $W$. Additionally, since the kernel function is analytical, its derivative can be calculated exactly. As a result of this, the SPH method is able to completely avoid making a finite difference approximation.

### 4.1.3 Interpolating Vector Derivatives

As it will appear in our equations describing the fluid, we must also have some manner by which we can calculate the divergence of a vector field defined on the mesh of particles. Similar to Equation 4.1, we may use the following identity for vector quantities,

$$\vec{f}(\vec{r}_a) = \int \vec{f}(\vec{r}_a)\delta(\vec{r}_a - \vec{r}_b)dV_b. \tag{4.7}$$

We then take the divergence of $\vec{f}(\vec{r}_a)$ with respect to $\vec{r}_a$ to obtain:

$$\nabla \cdot \vec{f}(\vec{r}_a) = \int \vec{f}(\vec{r}_b) \cdot \nabla W(|\vec{r}_a - \vec{r}_b|, h)dV_b + \mathcal{O}(h^2), \tag{4.8}$$

again noting that the only term that depends on $\vec{r}_a$ is the kernel, $W$. As with Equation 4.4, we discretize to obtain the following form of the SPH divergence operator:

$$\nabla \cdot \vec{f}(\vec{r}_a) \approx \sum_b \frac{m_b}{D_b^\star} \vec{f}(\vec{r}_b) \cdot \nabla W(|\vec{r}_a - \vec{r}_b|, h). \tag{4.9}$$

### 4.1.4 Kernel Functions

In this section, we discuss various properties of the kernel function, $W$, necessary to preserve both the accuracy and computational tractability of the SPH method.

Perhaps the most important property for kernel functions in SPH is that they are able to accurately represent a parcel of fluid of mass, $m$, over a range of values of the

smoothing parameter, $h$. To this end, we must choose a kernel that will reduce to a point-like value of $f(\vec{r})$ in the limit as $h \to 0$,

$$\lim_{h\to 0} f_{SPH}(\vec{r}) = \delta(\vec{r} - \vec{r'}) f(\vec{r'}) = f(\vec{r}). \tag{4.10}$$

Equivalently, we require that

$$\lim_{h\to 0} W(|\vec{r} - \vec{r'}|, h) = \delta(\vec{r} - \vec{r'}). \tag{4.11}$$

In this thesis, we restrict ourselves to kernel functions that are purely radial, i.e. they depend purely on on the magnitude of $|\vec{r} - \vec{r'}|$ and the smoothing length, $h$. As previously stated, we must also ensure that the kernel function is properly normalized to preserve the conservation of the fluid parcel's total mass $m$,

$$\int W(|\vec{r} - \vec{r'}|, h) dV' = 1. \tag{4.12}$$

A simple Gaussian would be a reasonable choice since it meets these requirements. However, this choice isn't suitable in practice because the Gaussian kernel is nonzero for all values of $\vec{r}$. Thus, each particle would be affected by (and affect) all other particles in the simulation domain. Numerically, this results in a RHS calculation that requires $\mathcal{O}(N^2)$ operations per evaluation, where $N$ is the total number of particles. In order to avoid this, we impose an additional constraint on the kernel function, namely being radially compact,

$$W(|\vec{r} - \vec{r'}| > R_{\text{support}}, h) = 0, \tag{4.13}$$

where we have introduced $R_{\text{support}}$ to be some finite radius at which kernel function is exactly zero. This constraint allows us to restrict each particles interactions to just $N_{\text{neighbors}}$ neighbors within a the sphere of radius $R_{\text{support}}$. Now, by using an efficient nearest neighbor search algorithm (See Section 6.1) to find all the particles within a sphere of radius $R_{\text{support}}$, we are able to reduce the $\mathcal{O}(N^2)$ interactions down to $\mathcal{O}(N_{\text{neighbors}} N)$ operations.

In order to preserve the symmetry in the RHS and ensure the second order accuracy of our method, we must also require that the kernel be symmetric about the positions $\vec{r}$ and $\vec{r}'$.

$$W(\vec{r}-\vec{r}',h) = W(|\vec{r}-\vec{r}'|,h). \tag{4.14}$$

In this thesis, we use two types of kernel functions: B-splines and Wendland functions.

#### *4.1.4.1 B-Splines*

The most commonly used kernel functions in SPH are B-spline functions [64]. These functions are defined as the one dimensional Fourier transforms of the form:

$$W(q,h) = \frac{C}{h^{\mathcal{D}}} b_n(q), \tag{4.15}$$

$$b_n(q) \equiv \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\frac{\sin(k/n)}{k/n}\right)^n \cos(kq)\mathrm{d}k, \tag{4.16}$$

where we have defined $q \equiv |\vec{r}-\vec{r}'|/h$ for convenience. The most commonly used B-spline is the first twice differentiable form, $n = 4$. This function is referred to as the "cubic spline" and takes the form:

$$W(q) = \frac{1}{\pi h^3}\begin{cases} \frac{1}{4}(2-q)^3 - (1-q)^3, & 0 \le q \le 1 \\ \frac{1}{4}(2-q)^3, & 1 < q \le 2 \\ 0. & q > 2 \end{cases} \tag{4.17}$$

#### *4.1.4.2 Wendland functions*

Another family of viable kernel functions are the Wendland functions [77]. For some time, these functions have found use as high quality and compact interpolation functions but have only recently been found to have highly desirable properties for SPH. This family of functions has been found to maintain highly ordered particle distributions, thus greatly boosting accuracy in dynamical SPH tests [58]. This is because these functions show a high level of resistance to the pairing instability that has been

known to plague SPH simulations. Linear stability analysis has shown [21] that a necessary condition for stability against particle pairing is that the multidimensional Fourier transform of the kernel is nonnegative. This condition combined with a finite support radius are the defining properties of the Wendland functions [77]. These functions are of the form:

$$W(q,h) = \frac{C}{h^{\mathcal{D}}}\psi_{lk}(q) \tag{4.18}$$

$$\psi_{lk}(q) \equiv \mathcal{I}^k(1-r)_+^l \tag{4.19}$$

where $(\cdot)_+ \equiv \max(0,\cdot)$, and the operator

$$\mathcal{I}[f(q)] \equiv \int_q^\infty sf(s)\mathrm{d}s. \tag{4.20}$$

In $\mathcal{D}$ dimensions, the functions $\psi_{lk}$, with $l = k+1+\lfloor \mathcal{D}/2 \rfloor$, have the desirable property that they have positive Fourier transforms. For our purposes, the $\psi_{4,3}$ and $\psi_{5,3}$ make excellent choices for our kernel in one and two/three dimensions, respectively. These Wendland functions take the following normalized forms:

$$W(q,h)_{1D} = \frac{55}{32h}(1-q)_+^7(1+7q+19q^2+21q^3), \tag{4.21}$$

$$W(q,h)_{2D} = \frac{78}{7\pi h^2}(1-q)_+^8(1+8q+25q^2+32q^3), \tag{4.22}$$

$$W(q,h)_{3D} = \frac{1365}{64\pi h^3}(1-q)_+^8(1+8q+25q^2+32q^3), \tag{4.23}$$

where we have once again used $q \equiv |\vec{r}-\vec{r}'|/h$.

### 4.1.5 Integral Approximation of Derivatives

An integral based expression for the second derivative in SPH was first used by Brookshaw in 1985 [14]. Until recently, this method had only been used for second derivatives, however we will apply this method to the first derivatives of our SPH formalism (namely the gradient). To this end, we follow the method presented in [27] and tested throughly in both [15] and [58]. This approximation has been shown

to provide up to ten orders of magnitude more accuracy than the traditional SPH approximation (Equation 4.6) when estimating the gradient of a well-ordered particle distribution [58].

To begin, we first define the following function for any differentiable function, $f(\vec{r})$,

$$\vec{I}_f(\vec{r}) \equiv \int [f(\vec{r}') - f(\vec{r})](\vec{r}' - \vec{r}) W(|\vec{r} - \vec{r}'|, h) \mathrm{d}V'. \tag{4.24}$$

Then, by using the Taylor expansion about $\vec{r}$,

$$f(\vec{r}') - f(\vec{r}) = \nabla f \cdot (\vec{r}' - \vec{r}) + \mathcal{O}(f''), \tag{4.25}$$

we can rewrite $I$ as:

$$I^i_f(\vec{r}) = \int [(\nabla f)^k_{|\vec{r}}(\vec{r}' - \vec{r})^k](\vec{r}' - \vec{r})^i W(|\vec{r} - \vec{r}'|, h) \mathrm{d}V' + \mathcal{O}(f''). \tag{4.26}$$

We will now define the matrix $\tilde{\tau}$ as,

$$\tilde{\tau}^{ki}(\vec{r}) = \int (\vec{r}' - \vec{r})^k (\vec{r}' - \vec{r})^i W(|\vec{r} - \vec{r}'|, h) \mathrm{d}V'. \tag{4.27}$$

which will allow us to express Equation 4.26 as

$$I^i_f(\vec{r}) = (\nabla f)^k(\vec{r}) \tilde{\tau}^{ki} + \mathcal{O}(f''). \tag{4.28}$$

Therefore the components of the gradient may be written as,

$$(\nabla f)^k(\vec{r}) = \tilde{\tau}^{-1,ki}(\vec{r}) \tilde{I}^i_f(\vec{r}) = \tilde{C}^{ki}(\vec{r}) \tilde{I}^i_f(\vec{r}). \tag{4.29}$$

where $\tilde{C} \equiv \tilde{\tau}^{-1}$.

We can see that $\tau$ contains only information about particle positions, while $I$ contains the information about the function to be approximated. By applying the SPH summation approximation (Equation 4.4) to the components of the matrix $\tau$, we arrive at the form:

$$\tau^{ki}(\vec{r}) = \sum_b \frac{m_b}{D^\star_b} (\vec{r}_b - \vec{r})^k (\vec{r}_b - \vec{r})^i W(|\vec{r} - \vec{r}_b|, h), \tag{4.30}$$

and

$$\left(\vec{I}_f(\vec{r})\right)_{IAD} = \sum_b \frac{m_b}{D_b^\star} f_b(\vec{r}_b - \vec{r}) W(|\vec{r} - \vec{r}_b|, h). \tag{4.31}$$

Where we have adopted in the conventions of [27] to label our "integral approximation of the derivative" as IAD. We now see that our approximation has the form:

$$\begin{aligned}(\nabla f)^k_{IAD}(\vec{r}) &= C^{kd}(\vec{r}) \left(I^d_f(\vec{r})\right)_{IAD} \\ &= \sum_b \frac{m_b}{D_b^\star} f_b \sum_{d=1}^{\mathcal{D}} C^{kd}(\vec{r}, h)(\vec{r}_b - \vec{r})^d W(|\vec{r} - \vec{r}_b|, h).\end{aligned} \tag{4.32}$$

It is apparent that the second summation of Equation 4.32 performs the same function as the traditional kernel gradient:

$$\nabla^k W(|\vec{r} - \vec{r}_b|, h) \to \vec{G} = \sum_{d=1}^{\mathcal{D}} C^{kd}(\vec{r}, h)(\vec{r}_b - \vec{r})^d W(|\vec{r} - \vec{r}_b|, h). \tag{4.33}$$

As shown in Chapter 5 we will substitute all kernel gradients with with $\vec{G}$ as defined in Equation 4.33.

# CHAPTER V

# GENERAL RELATIVISTIC SMOOTHED PARTICLE HYDRODYNAMICS

Having established a set of SPH equations with high accuracy and robustness, we will now derive the set of equations suitable for the SPH treatment of general relativistic hydrodynamics. In this chapter, we will arrive at the final form of our GRSPH equations by taking into account the modifications necessary in order to allow for a dynamical calculation of the smoothing length $h$, as well as to properly treat fluid shocks using an artificial viscous pressure.

## *5.1 Derivation of the GRSPH Evolution Equations*

Now that we have both a general relativistic treatment of Lagrangian hydrodynamics (Equations 3.59 - 3.61) and a means to discretize our equations using the integral based SPH method (Equations 4.4, 4.6, and 4.33), we will now arrive at the final form for our numerical evolution equations.

### 5.1.1 Relativistic Fluid Density

To begin, we need a suitable means to calculate the density of our fluid at each point on the mesh. As already shown in Equation 4.5, the SPH discretization of the continuity equation, Eq. 3.59, takes the form after the application of Equation 4.4:

$$D_a^\star = \sum_b m_b W(|\vec{r}_a - \vec{r}_b|, h_a). \tag{5.1}$$

Additionally, we must update the smoothing length, $h$ of each particle while updating the density. Using the method outlined in Section 5.1.1.1, ensuring that our

particles obey the following equation:

$$h_a(D_a^\star) = \eta \left( \frac{m_a}{D_a^\star} \right)^{1/\mathcal{D}}, \tag{5.2}$$

where $\mathcal{D}$ is the number of dimensions of the simulation and $\eta$ is a parameter of order unity.

#### *5.1.1.1 Adapting Resolution to Local Density*

Until this point, we have ignored the contributions to the SPH discretization that result from allowing the smoothing lengths to vary in time. In order to properly adapt the resolution to the fluid density, we must update the smoothing length based on the local density, $D^\star$ [51]. This is equivalent to making sure that at any given time each particle retains a roughly constant amount of mass within its sphere ($D_a^\star h_a^{\mathcal{D}} =$ const). This means that at each time step, the following equations must be solved self consistently:

$$D_a^\star(h_a) = \sum_b m_b W_{ab}(h_a), \tag{5.3}$$

$$h_a(D_a^\star) = \eta \left( \frac{m_a}{D_a^\star} \right)^{1/D}, \tag{5.4}$$

where $\eta$ is a parameter of order unity that can be show to be proportional to the number of neighbors used in the summations over neighboring particles. We generally use $\eta = 1.2 - 1.5$ (or 2.2 for Wendland kernels). Since $D^\star$ depends on $h$ and vice versa, we must find the root of the equation

$$f(\bar{h}) = h[D^\star(\bar{h})] - \bar{h} = 0 \tag{5.5}$$

The changes in $\{D^\star, h\}$ are generally very small between time iterations, so the root of this equation can usually be found in one or two Newton-Raphson iterations.

Since this update to the smoothing lengths changes the overall mass distribution, we must also introduce corrective terms to the evolution equations. To this end, we

define $\Omega$ as a corrective factor ($\mathcal{O}(1)$) resulting from the changing $h$:

$$\Omega_a = 1 - \frac{\partial h_a}{\partial D_a^\star} \sum_b m_b \frac{\partial W_{ab}(h_a)}{\partial h_a}. \tag{5.6}$$

The origin of this term is detailed in Appendix A.1. These are often referred to as as the "grad-h" terms in the literature [51]. Additionally, it can be shown that

$$\frac{\partial h_a}{\partial D_a^\star} = \frac{-h_a}{\mathcal{D} D_a^\star}, \tag{5.7}$$

with $\mathcal{D}$ being the number of dimensions. Therefore, Equation 5.6 takes the form,

$$\Omega_a = 1 + \frac{h_a}{\mathcal{D} D_a^\star} \sum_b m_b \frac{\partial W(r_{ab}, h_a)}{\partial h_a}. \tag{5.8}$$

While one could calculate the term $\partial W / \partial h_a$ directly from the definition of the kernel, it is also worth noting that we can calculate it as,

$$\frac{\partial W_{ab}(h_a)}{\partial h_a} = -\frac{|r_{ab}|}{h_a} \partial_a W_{ab}(h_a) - \frac{\mathcal{D}}{h_a} W_{ab}(h_a) \tag{5.9}$$

using the values of $W_{ab}(h_a)$ and $\partial_a W_{ab}(h_a)$ that are already necessarily calculated for each pair of particles.

### 5.1.2 The Momentum Evolution Equations

We will now derive our momentum evolution equation using the variational method first outlined in [70]. We will start by finding the discrete form of the Lagrangian for a general relativistic fluid [25],

$$\begin{aligned} L &= -\int T^{\mu\nu} U_\mu U_\nu \sqrt{-g} dV \\ &= -\int \rho_0 (1+u) \alpha \sqrt{\gamma} dV \\ &= -\int \frac{\alpha (1+u)}{W} D^\star \mathrm{d}V. \end{aligned} \tag{5.10}$$

By approximating our volume element as $\mathrm{d}V \approx \Delta V = m / D^\star$, and applying the same discretization as Equation 4.4, we arrive at the following form for the Lagrangian in

the SPH approximation,

$$L \approx -\sum_b m_b \frac{\alpha_b(1+u_b)}{W_b}. \tag{5.11}$$

We may now use the Euler-Lagrange equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial v^i}\right) - \frac{\partial L}{\partial x^i} = 0, \tag{5.12}$$

to find an appropriately symmetric version of our evolution equation while, at the same time, accounting for the variable smoothing lengths. For the full derivation of the momentum equation, see Appendix A.1.1. Using this method, we arrive at the following form for the evolution of our relativistic momentum, $S_i$:

$$\begin{aligned}\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} = -\sum_b m_b \Bigg[ &\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_a \partial_a W_{ab}(h_a) + \left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b \partial_a W_{ab}(h_b)\Bigg] \\ &- \left[E\partial_i\alpha - S_j\partial_i\beta^j - \frac{\alpha}{2}S^{jk}\partial_i\gamma_{jk}\right]_a .\end{aligned} \tag{5.13}$$

This form is very similar to what would be found by simply applying Equation 4.6 to the continuum Equation 3.60. The only difference between the results of these two derivation methods is the presence of kernels of two different smoothing lengths in the variational approach. As shown in [70], the SPH equations of this form introduce significantly less noise than is introduced by simply applying Equation 4.6 to the spatial derivatives and using the mean kernel ($W_{ab}(h) = \frac{1}{2}[W_{ab}(h_a) + W_{ab}(h_b)]$) to symmetrize the pressure forces.

Next we will follow the procedure for integral based derivatives outlined in Section 4.1.5. First, we define the vectors $G_a$ and $G_b$ that will act as a stand in replacement for the radial derivative of the kernel function, $\partial_a W_{ab}$:

$$G_a^k = \sum_{d=1}^{\mathcal{D}} C^{kd}(\vec{r}_a, h_a)(\vec{r}_b - \vec{r}_a)^d W(|\vec{r}_a - \vec{r}_b|, h_a) \tag{5.14}$$

$$G_b^k = \sum_{d=1}^{\mathcal{D}} C^{kd}(\vec{r}_b, h_b)(\vec{r}_b - \vec{r}_a)^d W(|\vec{r}_a - \vec{r}_b|, h_b) \tag{5.15}$$

We may now make the substitution $\partial_a W_{ab}(h_c) \to \vec{G}_c$ (Equation 4.33) to arrive at the following form for the time derivative of our relativistic momentum, $S_i$:

$$\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} = -\sum_b m_b \left[ \left( \frac{\alpha\sqrt{\gamma}P}{D^{\star 2}} \right)_a \vec{G}_a + \left( \frac{\alpha\sqrt{\gamma}P}{D^{\star 2}} \right)_b \vec{G}_b \right] - \left[ E\partial_i \alpha - S_j \partial_i \beta^j - \frac{\alpha}{2} S^{jk} \partial_i \gamma_{jk} \right]_a . \tag{5.16}$$

### 5.1.3 The Energy Evolution Equation

Similarly, as derived in Appendix A.1.2, the Lagrangian derivation of the SPH equations yields the following for the time evolution of our energy variable, $\bar{E}$:

$$\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t} = -\sum_b m_b \left[ \left( \frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}} \right)_a v_b^i \cdot \partial_a W_{ab}(h_a) + \left( \frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}} \right)_b v_a^i \cdot \partial_a W_{ab}(h_b) \right] + \left[ E\partial_t \alpha - S_j \partial_t \beta^j - \frac{\alpha}{2} S^{jk} \partial_t \gamma_{jk} \right]_a . \tag{5.17}$$

Once again, by substituting the integral based derivatives as $\partial_a W_{ab} \to \vec{G}$, where $\vec{G}$ is defined as in Equations 5.14 and 5.15, we arrive at the following form of our energy evolution equation:

$$\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t} = -\sum_b m_b \left[ \left( \frac{\alpha\sqrt{\gamma}P}{D^{\star 2}} \right)_a v_b^i \cdot \vec{G}_a + \left( \frac{\alpha\sqrt{\gamma}P}{D^{\star 2}} \right)_b v_a^i \cdot \vec{G}_b \right] + \left[ E\partial_t \alpha - S_j \partial_t \beta^j - \frac{\alpha}{2} S^{jk} \partial_t \gamma_{jk} \right]_a . \tag{5.18}$$

### 5.1.4 Artificial Viscous Pressure

While Equations 5.1, 5.16, 5.18, and an equation of state represent a complete set of equations necessary to describe a general relativistic fluid, they will prove intractable in presence of discontinuities [46]. Perfectly smooth initial conditions can often result in such discontinuities, known as shocks [39]. Due to the nature of fluid flows in the presence of compact objects, shocks are nearly ubiquitous in the astrophysical flows we aim to study. Numerically, shocks are often handled in one of two ways, either

by solving an analytical Riemann problem or by introducing artificial dissipation in order to spread out the discontinuous gradients over a resolvable scale. Guided by the latter method, first developed by Von Neumann and Richtmyer [76], we will add the required dissipation in the form of an additional pressure, $q$. By making the ad-hoc substitution, $P \rightarrow P + q$, in our evolution equations, we will smooth out the discontinuities by preventing the pile up of particles in regions that are being shocked. Such an over accumulation of particles in the presence of the shock would lead to nonphysical results and negatively affect the stability of our method. Von Neumann and Richtmyer began by laying out the following requirements for the artificial viscosity (AV):

1. In the presence of AV, no real discontinuities should occur.
2. The discontinuities should be spread over a "shock layer", with thickness on the order of a resolvable length scale, $l$.
3. The AV should have no noticeable effect in regions away from the shock layer.
4. The Rankine-Hugoniot conditions [54, 36] hold over a length scale that is large in comparison to the shock layer's thickness $l$. These conditions provide a set of constraints on the fluid on both sides of the shock front; constraints that arise from the various conservation properties of the fluid.

In order to meet these requirements, they proposed a form of:

$$q_{\mathrm{NR}} = \tilde{\beta}\rho_0 l^2 (\nabla \cdot \vec{v})^2, \tag{5.19}$$

where $\tilde{\beta}$ is a dimensionless parameter of order unity. Artificial viscosity of this form gives good results in the region of a shock, but has been shown to allow unphysical oscillations in the post shock region. In order to prevent these spurious oscillations one may introduce an additional term representing a bulk viscosity [40]:

$$q_{\mathrm{bulk}} = -\tilde{\alpha}\rho_0 c_s l (\nabla \cdot \vec{v}). \tag{5.20}$$

With these in mind, we will use the following form for the artificial viscous pressure acting on a particle $a$ as a result of its interaction with particle $b$ in a relativistic context:

$$q_{ab} = \begin{cases} \rho_{0,a}\omega_a \left( -\tilde{\alpha}_a c_{s,a} h_a (\nabla \cdot \vec{v})_{ab} + \tilde{\beta}_a h_a^2 (\nabla \cdot \vec{v})_{ab}^2 \right) & \text{for } (\nabla \cdot \vec{v})_{ab} < 0 \\ 0 & \text{for } (\nabla \cdot \vec{v})_{ab} \geq 0 \end{cases} \tag{5.21}$$

In Equation 5.21 we have chosen the scale length, $l$, to be the smoothing length, $h$, of our SPH particle and have set this artificial pressure to zero in regions that are not undergoing compression. We have also followed the strategy of [66] and multiplied the Newtonian $q$ by the enthalpy $\omega$ so that the values of $\tilde{\alpha}$ and $\tilde{\beta}$ may still be chosen to be of order unity even for ultra relativistic flows. In Equation 5.21, we will approximate the divergence of velocity between particles $a$ and $b$ as,

$$(\nabla \cdot \vec{v})_{ab} \approx \frac{\vec{v}_{ab} \cdot \vec{r}_{ab}}{|r_{ab}|^2 + \bar{\epsilon} h_{ab}^2}. \tag{5.22}$$

where $\bar{\epsilon}$ is a small numerical parameter chosen to avoid singular values of the divergence. Typically, we set this parameter to $\bar{\epsilon} = 0.1$.

In order to account for this pressure in the stress energy tensor of our particle, we will use the following equation to calculate the value of $q_a$ defined at the location of particle $a$,

$$q_a = \frac{1}{2} \sum_b m_b \left( \frac{q_{ab}}{D_a^\star} + \frac{q_{ba}}{D_b^\star} \right) W(|r_{ab}|, h_{ab}). \tag{5.23}$$

By introducing this artificial pressure into our evolution equations, Equations 5.16 and 5.18 will now take the form:

$$\begin{aligned} \frac{\mathrm{d}S_i}{\mathrm{d}t} =& -\sum_b m_b \left[ \left( \frac{\alpha\sqrt{\gamma}(P + q_{ab})}{D^{\star 2}} \right)_a \vec{G}_a + \left( \frac{\alpha\sqrt{\gamma}(P + q_{ba})}{D^{\star 2}} \right)_b \vec{G}_b \right] \\ & - \left[ \tilde{E}\partial_i \alpha - S_j \partial_i \beta^j - \frac{\alpha}{2} \tilde{S}^{jk} \partial_i \gamma_{jk} \right]_a \end{aligned} \tag{5.24}$$

$$\begin{aligned} \frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t} =& -\sum_b m_b \left[ \left( \frac{\alpha\sqrt{\gamma}(P + q_{ab})}{D^{\star 2}} \right)_a v_a^i \cdot \vec{G}_a + \left( \frac{\alpha\sqrt{\gamma}(P + q_{ba})}{D^{\star 2}} \right)_b v_b^i \cdot \vec{G}_b \right] \\ & + \left[ \tilde{E}\partial_t \alpha - S_j \partial_t \beta^j - \frac{\alpha}{2} \tilde{S}^{jk} \partial_t \gamma_{jk} \right]_a \end{aligned} \tag{5.25}$$

where we have replaced the fluid variables, $E$ and $S^{ij}$, with new versions which incorporate the substitution $P \to P + q$ into the stress energy tensor $\tilde{T}_{\mu\nu}$, i.e.:

$$\tilde{T}_{\mu\nu} = (\rho + P + q)U_\mu U_\nu + (P + q)g_{\mu\nu} \tag{5.26}$$

$$\tilde{E} = n^\mu n^\nu T_{\mu\nu}/D = \omega W - (P + q)/D \tag{5.27}$$

$$\tilde{S}^{ij} = \gamma^{\mu i}\gamma^{\nu j}T_{\mu\nu}/D = \omega W V^i V^j + [(P + q)/D]\gamma^{ij} \tag{5.28}$$

#### *5.1.4.1 Dissipation Triggering*

Artificial viscosity is only needed in very specific circumstances as a means to resolve shocks. In spite of this, traditional SPH methods apply the artificial viscosity everywhere, adding unphysical viscosity when it is not needed. In order to minimize the amount of unnecessary and unphysical dissipation we apply to the fluid, we employ a dissipation trigger. As first suggested by Morris and Monaghan [50], we will evolve the artificial viscosity parameter $\tilde{\alpha}$ in time, raising it in the presence of shocks and otherwise letting it decay exponentially. Further improving on the dissipation trigger method, Cullen and Dehnen [20] noted that it is more effective to instantaneously raise the dissipation parameter to its desired value and only evolve its decay over time. This gives us the following equation for the time evolution of $\tilde{\alpha}$

$$\frac{\mathrm{d}\tilde{\alpha}_a}{\mathrm{d}t} = \frac{-\tilde{\alpha}_a}{\tau_a}, \tag{5.29}$$

where we have chosen the decay time scale to be of the form

$$\tau_a = \chi \frac{h_a}{c_{s,a}}. \tag{5.30}$$

The parameter $\chi$ in Equation 5.30 is chosen to be $\chi = 2$ in order to ensure a rapid decay of $\tilde{\alpha}$. We will make use of two distinct ways to calculate the desired dissipation. The first, $\tilde{\alpha}_{shock}$, will trigger in the presence of a shock, while the second, $\tilde{\alpha}_{noise}$, will trigger on the more mild velocity noise that is often present in the region behind a

shock front. At each timestep, we will calculate both triggers and pick our desired value of $\tilde{\alpha}$ as

$$\tilde{\alpha}_{a,des} = \max(\tilde{\alpha}_{shock}, \tilde{\alpha}_{noise}). \tag{5.31}$$

When $\tilde{\alpha}_{a,des}$ is greater than the current value of our dissipation parameter, we will instantaneously set $\tilde{\alpha}_a = \tilde{\alpha}_{a,des}$ and $\tilde{\beta} = 2\tilde{\alpha}$. Otherwise, $\tilde{\alpha}_a$ will undergo its normal decay as dictated by Equation 5.29.

To calculate the shock trigger, we will use the time derivative of the compression as a shock indicator [20],

$$A_{a,shock} = \max\left(-\frac{\mathrm{d}(\nabla \cdot \vec{v})_a}{\mathrm{d}t}, 0\right). \tag{5.32}$$

Using this indicator we can now set the desired value of our dissipation parameter as

$$\tilde{\alpha}_{a,shock} = \tilde{\alpha}_{max} \frac{A_{a,shock}}{A_{a,shock} + (c_{s,a}/h_a)^2}, \tag{5.33}$$

where we have introduced the parameter $\tilde{\alpha}_{max}$ to represent the maximum allowed value of our dissipation parameter.

We will also use a noise trigger for the artificial viscosity [58]. This noise trigger is not strictly necessary (unlike the shock trigger) but has been found to improve convergence in regions with excessive noise in the velocity. Velocity noise is common in post shock regions and is characterized by fluctuations in the sign of $\nabla \cdot \vec{v}$ as the mesh "rings".

To indicate regions of velocity noise, we can use the ratio,

$$\frac{S_{1,a}}{S_{2,a}} = \frac{\sum_b (\nabla \cdot \vec{v})_{b,IAD}}{\sum_b |\nabla \cdot \vec{v}|_{b,IAD}}, \tag{5.34}$$

which will deviate from $\pm 1$ in the presence of velocity noise. In order to use this ratio as a dissipation trigger, we will introduce the following noise indicator (akin to the shock indicator in Equation 5.32):

$$\mathcal{N}_a^{(1)} = \left| \frac{\tilde{S}_{1,a}}{S_{2,a}} - 1 \right|, \tag{5.35}$$

where

$$\tilde{S}_{1,a} = \begin{cases} -S_{1,a} & (\nabla \cdot \vec{v})_a < 0 \\ S_{1,a} & \text{otherwise} \end{cases} \tag{5.36}$$

From these we may now define the desired dissipation parameter as

$$\tilde{\alpha}^{(1)}_{a,noise} = \tilde{\alpha}_{max} \frac{\mathcal{N}^{(1)}_a}{\mathcal{N}^{(1)}_a + \mathcal{N}_{\text{noise}}}, \tag{5.37}$$

where the parameter $\mathcal{N}_{\text{noise}}$ has been determined by experimentation to produce good results for $\mathcal{N}_{\text{noise}} = 50$ [58]. This noise trigger does not take into account the scale of the compression or expansion but rather simply considers the signs of $\nabla \cdot \vec{v}$. To account for the relative strength of the noise, we will introduce a second noise trigger. To this end, we define the following quantities to represent the average values of the compression and expansion of a particles neighbors:

$$\mathcal{S}^+_a = \frac{1}{N^+} \sum_{b, \nabla \cdot \vec{v}_b > 0}^{N^+} W_b \nabla \cdot \vec{v}_b \tag{5.38}$$

$$\mathcal{S}^-_a = \frac{-1}{N^-} \sum_{b, \nabla \cdot \vec{v}_b < 0}^{N^-} W_b \nabla \cdot \vec{v}_b, \tag{5.39}$$

where $N^+$ and $N^-$ are the number of particles undergoing expansion and compression, respectively. We will now define the relevant noise indicator as

$$\mathcal{N}^{(2)}_a = \sqrt{\mathcal{S}^+_a \mathcal{S}^-_a}, \tag{5.40}$$

In order to use $\mathcal{N}^{(2)}$ to calculate a desired dissipation parameter, we will compare it to the natural scale of the SPH method, $c_s/h$, so that it will only take effect in regions where the magnitude of the velocity noise is comparable to this scale,

$$\tilde{\alpha}^{(2)}_{a,noise} = \tilde{\alpha}_{max} \frac{\mathcal{N}^{(2)}_a}{\mathcal{N}^{(2)}_a + 0.2(c_{s,a}/h_a)}. \tag{5.41}$$

Finally, we will take the value of the desired dissipation parameter resulting from the velocity noise to be the max of Equations 5.37 and 5.41:

$$\tilde{\alpha}_{a,noise} = \max(\tilde{\alpha}^{(1)}_{a,noise}, \tilde{\alpha}^{(2)}_{a,noise}). \tag{5.42}$$

# CHAPTER VI

# COMPUTATIONAL AND NUMERICAL METHODS

In this chapter we will discuss the various computational and numerical methods used to implement our SPH method. In particular, we will discuss the spatial hashing method that we have used in order to perform a search for each particles relevant neighbors. We will then outline the methods used to share the computational load on the distributed computing systems that are the standard for modern high performance computers. We will end with a discussion on the time integration methods used and corresponding limits on our integrators' timestep.

## *6.1 Hierarchical Spatial Hashing*

The most computationally demanding step in any SPH code is the N-nearest neighbor search used to find each particle's neighbors. Traditionally, most SPH codes solve this problem with $\mathcal{O}(N \log N)$ complexity by making use of an octtree data structure [68]. Simply put, a standard octtree is a multiply linked list, where each node represents a portion of the spatial domain further subdivided by its subnodes. The children of a node then represent a subvolume of their parent with each parent having $2^{\mathcal{D}}$ children, where $\mathcal{D}$ is the dimensionality of the domain. In order to find a particle's neighbors in such a data structure, one must traverse the the octtree to find all spatial volumes that are of relevance to the particle at hand. Traversal of the tree structure in this manner has proven to be a slow prospect for modern computers as it is largely limited by the speed of the memory rather than the CPU's arithmetic capacity.

To alleviate the problems with a standard octtree, we approach this problem using a 'Hierarchical Spatial Hashing' based octree-like method. The basic idea of this method is to replace the linked list with a hash table. This eliminates the need

to traverse many memory pointers while querying the structure and replaces it with a trivial computation and a data structure that allows for quick random access. This method is not only more parallel, but shows massive serial performance benefits as well. By calculating a "cheap" hash function for each particle's position, we are able to completely avoid the slowest part of an octtree while still maintaining a similar layout for the data. With most modern CPUs being overprovisioned for arithmetic relative to the rate at which they can fetch data from memory, calculating a hash function will often come at little or no cost when compared to the time spent retrieving data to query a traditional octree.

### 6.1.1 Hashtable Construction

In order to construct a hashtable containing our particles, we use the following approach to calculate a hash function that is well distributed in key-space [13, 23, 65].

First we must map each particle's real valued position and "size" ($\{x^i, h\} \in \mathbb{R}^4$) to a set of four positive integers $\{i, j, k, l\} \in \mathbb{I}^{+4}$ as follows:

$$\begin{pmatrix} x \\ y \\ z \\ h \end{pmatrix} \rightarrow \begin{pmatrix} \lfloor x/2^l \rfloor \\ \lfloor y/2^l \rfloor \\ \lfloor z/2^l \rfloor \\ \lceil \log_2(2h) \rceil \end{pmatrix} = \begin{pmatrix} i \\ j \\ k \\ l \end{pmatrix},$$

where $\lfloor x \rfloor$ and $\lceil x \rceil$ are the floor and ceiling functions of $x$, respectively. The "bin size", represented by $l$, is chosen such that each particle's spatial influence will not extend beyond the immediately surrounding bins (see Figure 6.1).

We must then combine the integers $\{i, j, k, l\} \in \mathbb{I}^{+4}$ into one integer index value $H \in [0, m-1] \in \mathbb{I}^{+}$ by using a hash function $\mathcal{H}(i, j, k, l)$, where $m$ is the size of our hashtable. The hash function, $\mathcal{H}$, should have the following very important properties:

- This function should distribute hashes as uniformly as possible over the whole space $[0, m-1]$ to minimize collisions in the hashtable. This is critical to prevent

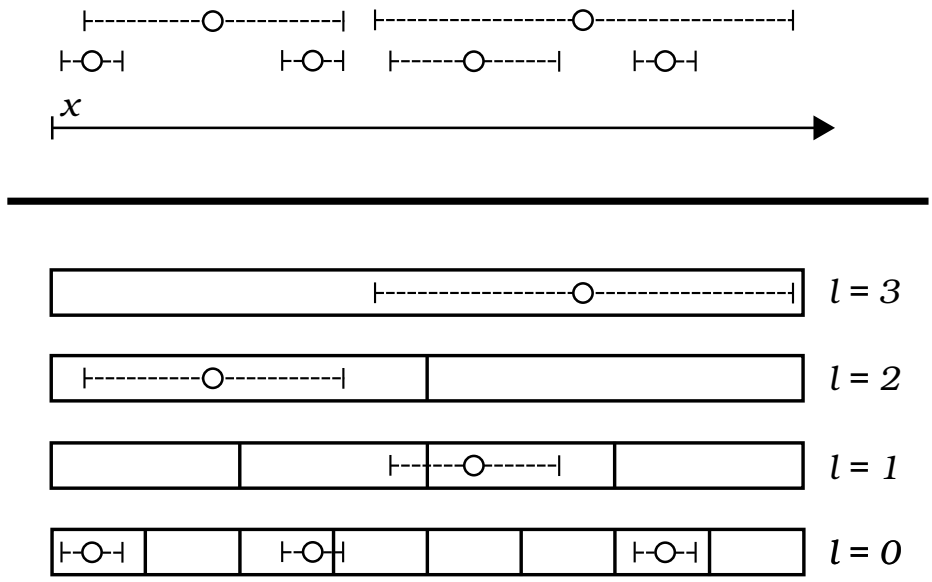



Figure 6.1: Hash table insertion: For the example distribution of finite sized particles at the top, the particles would be inserted into the hash table as shown. First the value of $l$ would be calculated for each particle, then the particle would be inserted into the $i$th bin, where $i = \lfloor x/2^l \rfloor$

hash collisions that will drastically slow down our query by forcing us to perform unnecessary work.

- The CPU should able to calculate the function, $\mathcal{H}$, using very few clock cycles as it will be called very many times both during the construction of the hash table and its subsequent queries. To this end, we want the function $\mathcal{H}$ to include only "easy" binary arithmetic (e.g. multiplication, XOR, and mod) while avoiding more complex operations (e.g. division, exponentiation).

With these criteria in mind, we find that the following hash function is well suited for our needs:

$$\mathcal{H}(i, j, k, l) = (i \times p_1 \oplus j \times p_2 \oplus j \times p_3 \oplus l \times p_4) \mod m, \tag{6.1}$$

where $p_1, p_2, p_3, p_4$ are large prime numbers[1], $\oplus$ is binary XOR, and the table size,

[1]We use $\{p_i\} = \{73856093, 19349663, 83492791, 67867979\}$ [23]

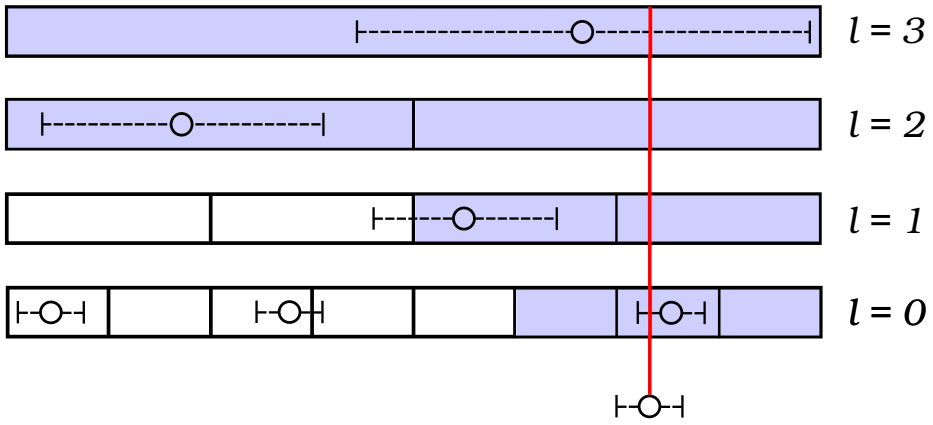



Figure 6.2: Hash table query: For the example hash table built in Figure 6.1, a query for the neighbors of the particle at the bottom would require searching all the bins marked as blue. The red line represents our query particle's position in each $l$ level from which we must query all bins within the range $[i-1, i+1]$.

$m$, is taken to be the next prime that is larger than half of the maximum value of the implementations unsigned integer [72].

### 6.1.2 Nearest Neighbor Search

As a result of the hashtable's structure, querying for nearest neighbors is rather easy. Put simply, to query for the neighbors of a particle at $\vec{r_a}$ we need to find all the particles such that

$$|\vec{r_a} - \vec{r_b}| \leq h_{nn}, \tag{6.2}$$

where $h_{nn} = \max\{h_a, h_b\}$ is the maximum smoothing length among particles $a$ and $b$. In order to do this, we must query the hash table by checking each key in the range $\{i_a \pm 1, j_a \pm 1, k_a \pm 1\}$ for each value of $l$ present in the table (Figure 6.2). This amounts to 3, 9, or 27 queries per particle per $l$ in 1, 2, and 3 dimensions, respectively. By querying each possible value of $l$ associated with the point, $r_a$, and its neighbors, we are assured that our list of potential neighbors will contain all particles for which Equation 6.2 could be true. We then check each particle returned by the query against

Equation 6.2 to build our final list of neighbors[2].

### 6.1.3 Nearest Neighbor Search Scaling

To test the algorithmic complexity of the hierarchical spatial hashing method, we generate a distribution that is expected to be difficult to query for nearest neighbors in that it spans many orders of magnitude in local particle density. For this, we randomly distribute particles according to a radially Gaussian probability density function in three dimensions:

$$P(r) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{r^2}{2\sigma^2}). \tag{6.3}$$

Figure 6.3 shows the scaling of the wallclock time needed to calculate the complete neighbor list for each of N particles in such a distribution. As we expected, this method is well fit by a curve of the form $N^{1.068}$, implying a computational complexity of approximately $\mathcal{O}(N)$ even for a system that spans many orders of magnitude in particle size.

## *6.2 Exploiting the Parallelism of the SPH Method*

In order to fully utilize modern supercomputers, we must be able to exploit various levels of parallelism to speed up our code. By adding more computational power, we hope to reduce our run time proportionally. The speedup that one can expect to achieve is referred to as Amdahl's Law [5]:

$$S_{latency}(s) = \frac{1}{(1-p) + p/s + Os}, \tag{6.4}$$

where $S_{latency}$ is our expected speedup as a function of the theoretical speedup of a perfectly parallel portion of the code, $s$. The parameters of Equation 6.4, $p$ and $p_o$,

[2]It should be noted that this step is not strictly necessary because all particles not meeting the condition of Equation 6.2 will contribute exactly zero to the pairwise calculations in the right hand side of our evolution equations. However, in practice, the additional computation expended while building the list of neighbors more than compensates for the otherwise wasted effort spent while calculating the right hand side.

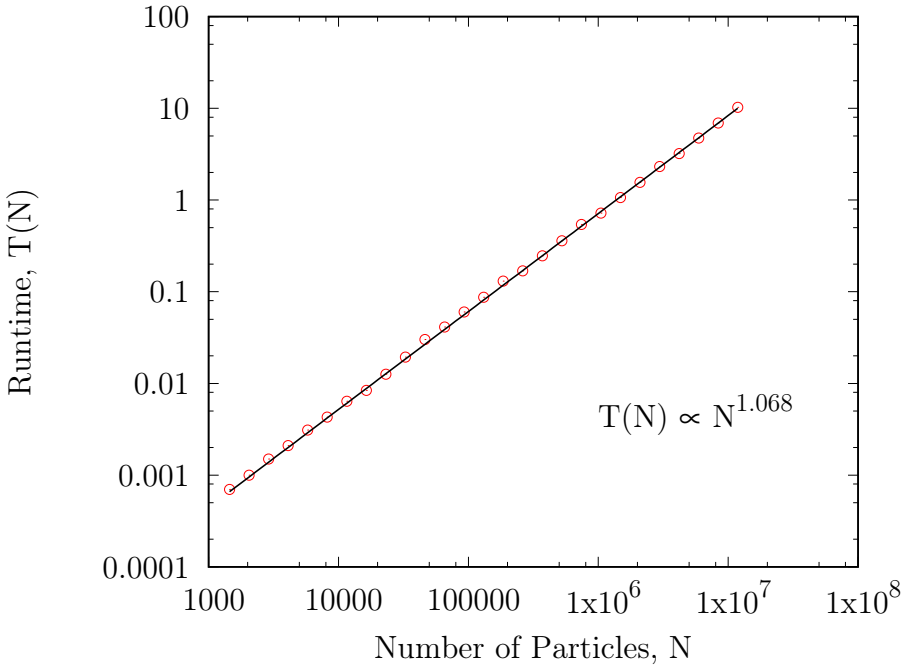



Figure 6.3: Nearest neighbor search scaling: wallclock time required to perform a nearest neighbor search for all N particles, randomly placed following a three dimensional radially Gaussian probability density function.

represent the fraction of the code that is parallel and the overhead incurred by adding more resources, respectively. In Amdahl's original work, the parameter $p_o$ is zero, but we find that it provides a good means to describe the overhead that often results from the communication that naturally results from the usage of more resources.

### 6.2.1 Shared Memory Parallelism

The nature of the SPH method makes shared memory parallelism (e.g. OpenMP, Intel TBB) relatively easy to obtain. In the simplest case, at each time step, we must calculate the right hand side of $6N$ ordinary differential equations. In practice, this manifests itself as a loop over the particle indices which contains a subloop over the indices of neighboring particles. The outer loop can be easily distributed to multiple threads using one of any number of shared memory libraries. In particular, we use either OpenMP or Intel's Thread Building Blocks (TBB) depending on our current system's architecture. We find that very little performance tuning is necessary at this level of parallelism when using a dynamic thread scheduler. Using dynamic scheduling, each thread is sent a small portion of the total particles as its workload. Once complete, the thread will ask for more work until the entire pool of available work has been exhausted. We favor this type of scheduling over the default 'static' scheduling due to the uneven nature of the workload.

Static scheduling looks at the total number of loop iterations that must be performed and evenly splits them among all available threads. When the workload per loop iteration is equal, this method is favorable because it minimizes overhead resulting from thread allocation. However this can lead to load balancing issues when the amount of work performed by each loop iteration is not uniform. In SPH, the lion's share of the work needed in order to calculate the RHS occurs inside the inner loop over each particle's neighbors. This presents a problem for static scheduling because

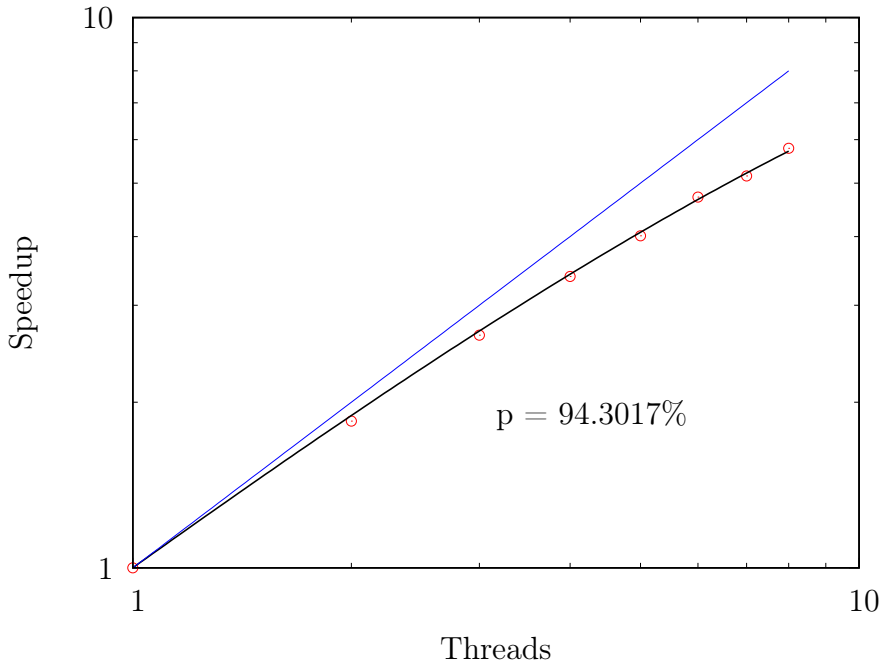



Figure 6.4: OpenMP scaling: The speedup of the average wallclock time for one hundred timesteps as a function of OpenMP threads, fit to Amdahl's law (Equation 6.4). The blue line represents perfect scaling. A similar scaling is observed when using Intel TBB.

the number of neighbors that each particle has is varied by up to a factor of approximately 2, leading to load balancing issues. As a result, the communication overhead that is incurred by dynamic scheduling is ameliorated since we are now able to keep all threads operating constantly. Figure 6.4 shows a fit of the speedup (averaged over 100 full timesteps) achieved with up to 8 shared memory threads on a 2.7 GHz Intel Xeon 8-Core 64-bit E5-processor.

### 6.2.2 Distributed Parallelism

The more complex issues of parallelism arise once we no longer restrict our particles to share the same memory space, as is common in distributed systems. For such systems, we must manage which processors own which particles while maintaining a relatively balanced load [5]. We achieve this by using the Message Passing Interface

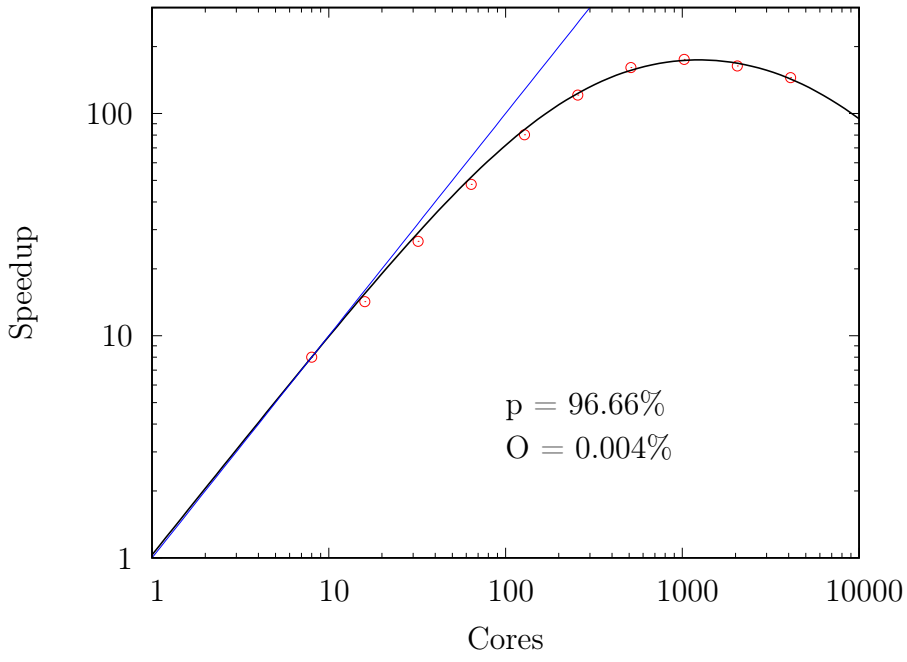



Figure 6.5: MPI scaling: The speedup averaged over one hundred timesteps as a function of the number of cores, fit to Amdahl's law (Equation 6.4). The blue line represents perfect scaling.

(MPI) to manage a set of processes that are distributed over the available processors. We bind one process to every physical processor and take advantage of shared memory parallelism to leverage each of the processor's available cores. This method is commonly referred to as MPI+OpenMP parallelism and is an excellent choice for the nonuniform memory access architectures that compose most modern supercomputers.

Since the forces acting on our particles arise from spatially local neighboring particles, it is also important that our method of distribution maintains as much spatial locality as is reasonably possible.

#### *6.2.2.1 Domain Decomposition*

The basis for our domain decomposition among MPI processes is simply that of Orthogonal Recursive Bisection [22, 9]. In ORB, we start by choosing a value $n$ such that our total number of MPI processes is $2^n$. In order to divide the domain among

these processes we repeatedly apply the following procedure, starting from the global extents of the domain and initially operating on all particles in the domain.

1. First calculate the dimension which has the largest span, i.e. $i$ such that $\max\{r^i_{max} - r^i_{min}\}$. We will divide the set of particles in two along this dimension.

2. Next, take all particles in the domain being split and calculate their mean[3] position along dimension to be split. At this position, we divide the particles into two groups.

3. Repeat these steps on each subdomain created in step 2. After we have recursively performed this split for all $n$ levels, we have completely decomposed our domain into $2^n$ blocks.

An example domain decomposition using this method is shown in Figure 6.6.

[3]In theory, using the mode would load balance more effectively. However, in practice, this would require the particles be sorted in the splitting dimension. Since this is an $\mathcal{O}(N \log N)$ procedure, we instead approximate it by using the mean.

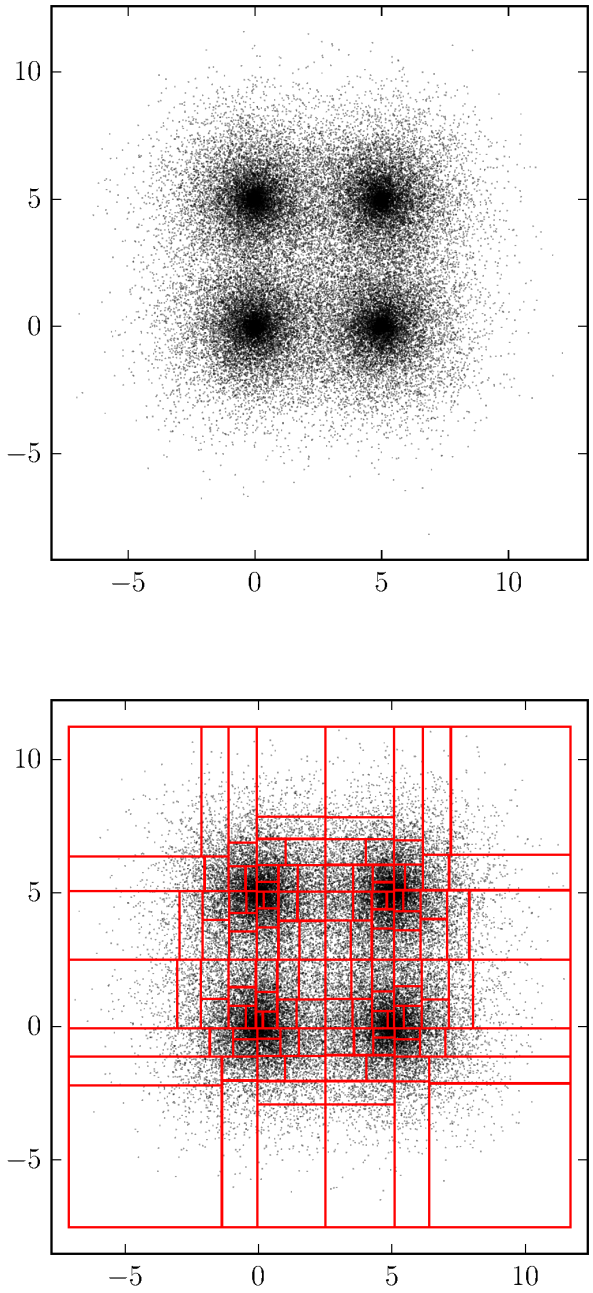



Figure 6.6: Orthogonal Recursive Bisection: For an example distribution (top), the ORB method produces the domain decomposition (bottom) for $2^8$ processors. Note that regions with sparse particle distributions have larger bounding boxes than those with high densities. This leads to a roughly constant number of particles in each bounding box.

#### *6.2.2.2 Load Balancing*

Another important aspect of designing a code for distributed computing is the problem of load balancing. Consider, for example, a situation where each process takes 0.8 seconds per iteration, with the exception of one that takes 1 full second. This presents a problem because no processes can continue until all particles have been updated to the same point. In such a scenario, the total time step will take 1 second, but for 20% of that time, most of the available processing capacity will sit idle. [9] Thus, in order to load balance our simulations, it is important to make sure the total amount of work is evenly divided among all available processors. Unfortunately, the amount of work required on a per particle basis is difficult to predict, so evenly dividing the particles among processors does not fully address this issue. However, since the simulations use a large number of particles, $N_p >> 1$, we are able to use timing statistics to work around this problem.

To begin, we simply time the evaluation of the RHS on each process, carefully excluding any blocking MPI functions that will result in each process waiting for the other to catch up. Since we know the structure of the domain decomposition from the ORB decomposition, we can then sum the total time taken by all processes on each side of the split made in each level in the ORB tree. With this knowledge, we then move the location of the split such that each side takes a comparable amount of time:

$$x^{\text{new}}_{\text{boundry}} = x^{\text{old}}_{\text{boundry}} + \Delta x_{\text{boundry}} \tag{6.5}$$

where

$$\Delta x_{\text{boundry}} = \begin{cases} K\Delta w & |K\Delta w| \leq r_{max}, \\ r_{max} & K\Delta w > r_{max}, \\ -r_{max} & K\Delta w < -r_{max}, \end{cases} \tag{6.6}$$

and $\Delta w$ represents the difference in fractional wait time $w = (t_{max} - t_{proc})/t_{max}$

between the two subvolumes on the positive ($_+$) and negative ($_-$) sides of the division,

$$\Delta w = w_- - w_+. \tag{6.7}$$

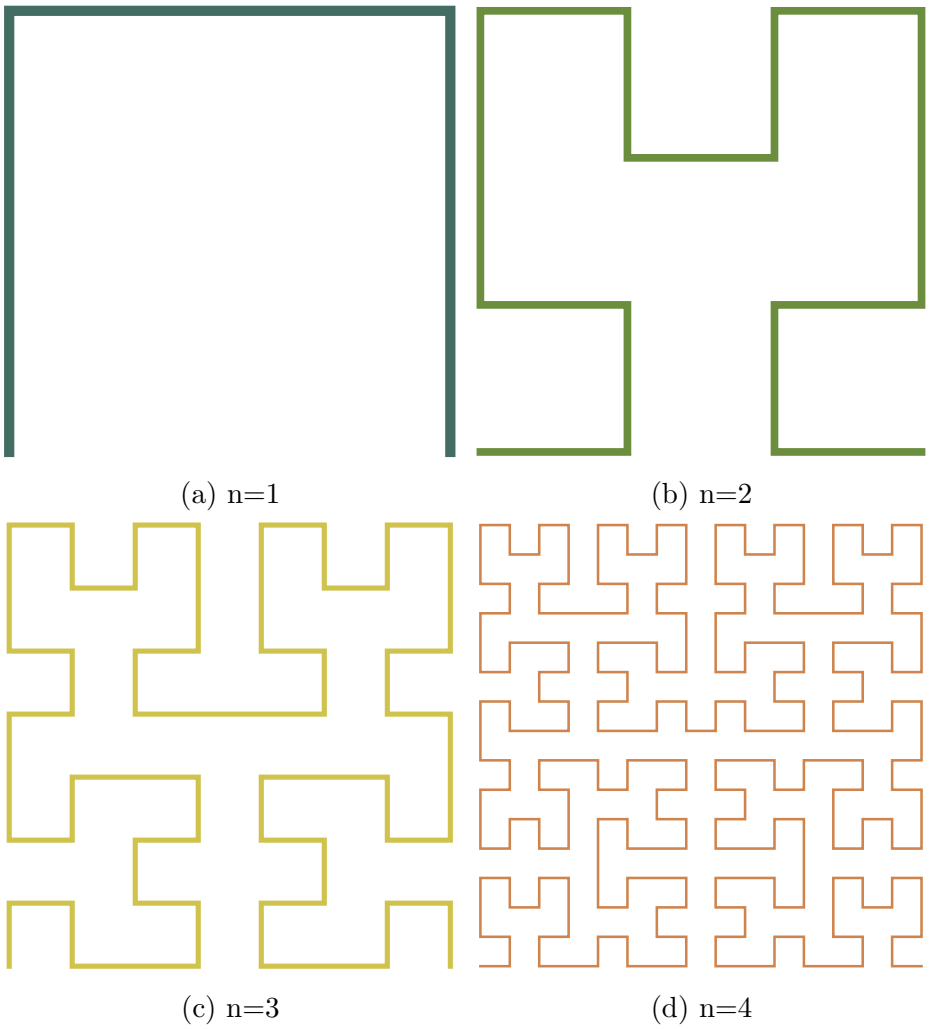
We have also introduced the parameters $K > 0$ and $r_{max} > 0$ to define a boundary "speed" and maximum correction for each time step. In practice, we find $K = \max\{w\}$ and $r_{max} = 2^{l_{min}}$ to provide a good balance of stability of the boundaries and a fast approach to a load balanced state.

## *6.3 Particle Ordering for Cache Locality*

Another important consideration is our ability to maintain memory locality for particles that will consistently interact [68]. By doing this, we can dramatically decrease the number of cache misses while calculating the RHS, leading to noticeable performance increases. To this end, we actively maintain each particle's location within the data arrays in order to maximize spatial locality. We use a fractal Peano-Hilbert space filling curve to map from the multi-dimensional space to a one dimensional index [49]. With this method we ensure that a pair of particles that are close together in the spatial domain have a high probability of also being close in array indices. Figure 6.7 shows various degrees of space filling curves of this type.

We periodically go through the following procedure to arrange our particle based on spatial locality. We first choose a fixed resolution with which we will bin our particles. This size is generally chosen to be the average value of $h$ among the particles local to each process. This value is generally a good choice as too large of a bin size would limit the spatial locality of our indices. Additionally, too small of a bin size would result in a much larger amount of computations to order the particles since they would be much more likely to change linear index between two sorts.

We next calculate the Peano-Hilbert key for all the particles in the local domain. This is done by casting each particle's position to a set of integer indices with respect to the bin size. With that information we calculate the Peano-Hilbert key over the



(a) n=1 (b) n=2

(c) n=3 (d) n=4

Figure 6.7: Two dimensional Hilbert curves of various degrees, n.

entire spatial domain, reducing each particles integer position to one integer value specifying its location on the curve.

Finally, we sort the particles in the order of their keys. Although the sorting of the Peano-Hilbert keys exhibits a worst case time complexity of $\mathcal{O}(N \log N)$, we find that, in practice, it can be done relatively quickly compared to the NN search or RHS calculations. This is because between calculations, the proper ordering of the indices changes only very slightly. This means that a well implemented sorting algorithm can achieve very close to its best case complexity of $\mathcal{O}(N)$ operations. The time that is saved through an increased rate of cache hits during the NN search and RHS

calculation is then significantly greater than the time spent performing the sort itself.

## 6.4 Time Integration

To integrate the particles forward in time, we employ one of multiple methods. We prefer to use a second or third order Total Variation Diminishing (TVD) Runge-Kutta (RK) method, over the simple "leapfrog" method that is common to many SPH implementations. In addition to their greater accuracy, these TVDRK methods maintain all of our state variables at the same time. While we have also implemented the more traditional "leapfrog" method, in which the velocity (and momentum $S_i$) is defined at half timesteps, we will not detail it here as it is not used in this work (See, for example, [42]).

The following subsections detail the methods that are used to generically evolve equations of the form:

$$\frac{d\vec{u}}{dt} = f(\vec{u}) \tag{6.8}$$

from the time $t^n$ to the time $t^{n+1}$ where the superscripts denote the time in integer units of $\Delta t$. In the implementation of our relativistic SPH formalism, the vector $\vec{u}$ will consist of the variables $r^i, S_i, \bar{E}_i$, as well as the dissipation trigger, $\tilde{\alpha}_i$.

### 6.4.1 Total Variation Diminishing Runge-Kutta Integration Methods

In our time integration scheme, we seek a method that will be highly stable in the presence of shocks. In this sense, we find that schemes exhibiting the Total Variation Diminishing (TVD) property, first outlined in [31], are an excellent choice. We use the following Runge Kutta methods that exhibit the TVD property:

1. The second order accurate ($\mathcal{O}(\Delta t^2)$ error) Total Variation Diminishing Runge-Kutta (TVDRK2) of the form [29]:

$$\vec{u}^{n+\frac{1}{2}} = \vec{u}^n + \Delta t f(\vec{u}^n) \tag{6.9}$$

$$\vec{u}^{n+1} = \frac{1}{2}\vec{u}^n + \frac{1}{2}(\vec{u}^{n+\frac{1}{2}} + \Delta t f(\vec{u}^{n+\frac{1}{2}})) \tag{6.10}$$

2. The third order accurate ($\mathcal{O}(\Delta t^3)$ error) Total Variation Diminishing Runge-Kutta (TVDRK3) of the form [29]:

$$\vec{u}^{n+\frac{1}{3}} = \vec{u}^n + \Delta t f(\vec{u}^n) \tag{6.11}$$

$$\vec{u}^{n+\frac{2}{3}} = \frac{3}{4}\vec{u}^n + \frac{1}{4}(\vec{u}^{n+\frac{1}{3}} + \Delta t f(\vec{u}^{n+\frac{1}{3}})) \tag{6.12}$$

$$\vec{u}^{n+1} = \frac{1}{3}\vec{u}^n + \frac{2}{3}(\vec{u}^{n+\frac{2}{3}} + \Delta t f(\vec{u}^{n+\frac{2}{3}})) \tag{6.13}$$

### 6.4.2 Timestep Limiting

For any numerical integrator, it is important to choose the size of the time step with care. A time step that is too large can easily lead to a loss of accuracy or even numerical instability.

#### *6.4.2.1 CFL Criteria*

To avoid such problems it is common to enforce the Courant-Friedrichs-Lewy (CFL) condition [19]:

$$\delta t_{\text{CFL}} \leq C_{\text{CFL}} \frac{\delta x}{c}, \tag{6.14}$$

where $\delta x$ is some characteristic length scale and $c$ is a characteristic speed. With our SPH formulation, we have a natural choice for each. The smoothing length, $h$, represents an appropriate length scale and the speed of sound, $c_s$, an appropriate speed. With this we arrive at the timestepping criterion:

$$\delta t_{\text{CFL}} \leq C_{\text{CFL}} \frac{h}{c_s}. \tag{6.15}$$

Physically, this plays the role of preventing the transfer of spatial information at a rate faster than the local sound speed. Rather conservatively, we choose a value of $C_{\text{CFL}} = 0.3$.

#### 6.4.2.2 Force Criteria

Additionally we enforce a timestep condition that is based on the acceleration of each particle. Specifically, we require that

$$\delta t_a \leq C_f \sqrt{\frac{h_{min}}{|d\vec{S}/dt|_{max}}}, \tag{6.16}$$

where $C_f$ is a parameter most often chosen to be in the range $0.25 - 0.5$. We have chosen $C_f = 0.3$.

#### 6.4.2.3 General Criteria

We also employ a generalized timestep criterion for other time dependent variables. By using simple dimensional analysis, we define a characteristic time scale for any time varying quantity, $x$:

$$\tau_x = \frac{x}{\dot{x}}. \tag{6.17}$$

In order to ensure that this timescale is properly resolved, we limit the timestep such that

$$\delta t_x = C_x \frac{x}{\dot{x}}, \tag{6.18}$$

where $C_x$ is again a parameter restricted to the range $0.25 - 0.5$, set to a value of 0.3 for our purposes.

In particular, we impose this criterion on the smoothing length ($x = h$) of each particle. Since we allow the smoothing length to vary in time based on the density (Section 5.2), it is important that we ensure a particle's size does not increase so quickly that it is a detriment to the stability of our method i.e.:

$$\delta t_h = C_h \frac{h}{\dot{h}}, \tag{6.19}$$

where we have, once again, chosen the constant $C_h$ to be 0.3.

#### *6.4.2.4 Setting the Timestep*

Having established a set of conditions (Equations 6.15, 6.16 and 6.19) on our timestep, we will satisfy all of them by requiring that the global timestep, $t_g$, is:

$$\delta t_g \leq \min\{\delta t_{CFL}, \delta t_a, \delta t_h\}. \tag{6.20}$$

# CHAPTER VII

# CODE VALIDATION

In this chapter, we will test our general relativistic SPH formalism against various standard test problems. These problems will include tests of our ability to correctly resolve fluid shocks as well as tests of the overall accuracy of our code in difficult scenarios. In particular, we will investigate our code's convergence properties for both the Newtonian and relativistic shock tube tests and the Gresho-Chan vortex test. We will show its ability to resolve a mildly relativistic blast wave as well as the Kelvin-Helmholtz instability. Finally, we will test our code in the non-flat spacetime of a Schwarzschild black hole using the Bondi solution describing steady state accretion onto a black hole.

## *7.1 One-Dimensional Sod Shock Tube*

Shocks are exceedingly common in astrophysical scenarios. In general, we most commonly see shocks in two different situations. In the first case, we have some amount of gas that is moving with higher velocity than another bit of gas with which it is in contact. When the higher velocity gas overtakes the slower gas, a discontinuity is formed in both density and velocity, resulting in a shock. Also common is the case where some portion of gas experiences supersonic compression. To test our ability to accurately resolve shocks we compare the code results to those of common analytic solutions involving shocks. The first test of this sort is that of the nonrelativistic Sod shock tube [67].

### 7.1.1 Setup

A classic test for hydrodynamics codes, the Sod shock tube consists of a removable membrane separating two regions of fluid, one on the left ($L$) and the other on the right ($R$). Each side of the fluid is initially at rest and has a unique pressure and density, e.g. $\{P, v_x, \rho_0\}_L = \{1, 0, 1\}$ and $\{P, v_x, \rho_0\}_R = \{.1, 0, .125\}$, chosen such that the speed of sound $c_s$ is constant across the discontinuity. These initial conditions result in a hydrodynamical solution that contains three important features: a shock wave, a contact discontinuity, and a rarefaction wave.

In order to test our code in the classical regime, we must rescale this test to ensure that our velocities remain low, $v << 1$. To do this, we simply multiply our distances and pressures by a factor of 0.001 while keeping our density constant. Having done this, we now have a system over the domain $[-0.0005, 0.0005]$ with initial conditions $\{\rho_R, \rho_L, P_R, P_L\} = \{1, 0.1, 0.001, 0.000125\}$ for a gas with an adiabatic index $\Gamma = 5/3$. In order to properly resolve the initial discontinuity, we will smooth the two states over some finite number of particles, $N_{\text{smooth}}$, at the interface using,

$$s(x) = \begin{cases} s_L & \text{if } d < -N_{\text{smooth}} \\ (s_L + s_R e^d)/(1 + e^d) & \text{if } |d| < N_{\text{smooth}} \\ s_R & \text{if } d > N_{\text{smooth}} \end{cases} \tag{7.1}$$

where s is one of our state variables ($\rho$,$P$) and $d$ is the distance from the interface in units of the separation between neighboring particles, $\Delta x$,

$$d = (x - x_{\text{interface}})/\Delta x. \tag{7.2}$$

For this test, we have chosen $N_{\text{smooth}} = 10$ as it is consistent with the average number of neighbors influencing each particle throughout the simulation.

The results of this test are shown in Figure 7.1. With a relatively high number of particles ($N = 3600$), we see excellent agreement with the analytical solution. At this

resolution the errors present are still visible to the eye. Namely, we see a slight "blip" in the pressure at the contact discontinuity. It is also clear that our shock trigger, $\tilde{\alpha}$, is working as expected, having its maximum value at the shock front and having only a moderate value in the regions of unphysical velocity noise.

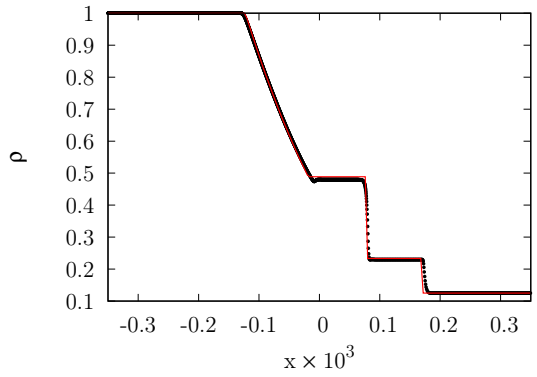
### 7.1.2 Convergence

In order to test the convergence of our code, we will use the $L_1$ norm when comparing our variables to their analytic solution. For the numerical variable, $x$, the $L_1$ norm takes the form:

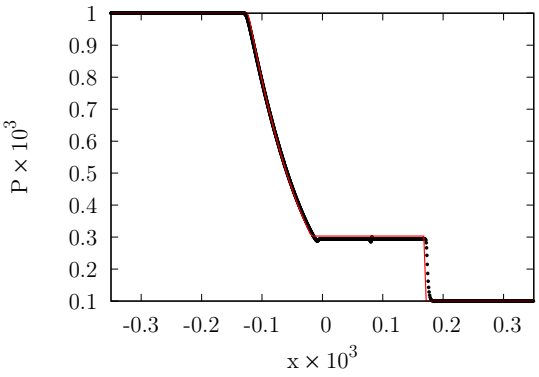
$$L_1(x) = \frac{1}{N} \sum_a |x_a - x_{\text{exact}}(\vec{r}_a)|, \tag{7.3}$$

where $N$ is the number of particles in the simulation domain, $x_a$ is the numerical value of $x$ at particle $a$, and $x_{\text{exact}}(\vec{r}_a)$ is the exact solution at the position of particle $a$. In our convergence tests, we will choose to calculate the $L_1$ error of our velocity, $L_1(v)$, as the $L_1$ norm of the density is subject to bias [69].

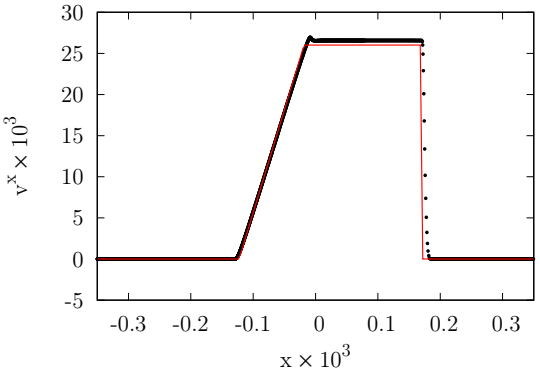
For this test, we expect an optimal convergence rate of $L_1(v) \propto N^{-1}$ due to the reduced order of our method in the presence of shocks. In Figure 7.2, we show the convergence of the $L_1$ error for the Sod type shock tube test. In this non-relativistic test, we find slightly suboptimal convergence of approximately $N^{-.866}$. This is likely the result of using our relativistic method on this non-relativistic problem. Even by rescaling the problem (as outlined in section 7.1.1), this test still results in the shock wave having a Lorentz factor of $W = 1.0045$, leading to slight disagreement with the exact Newtonian solution.



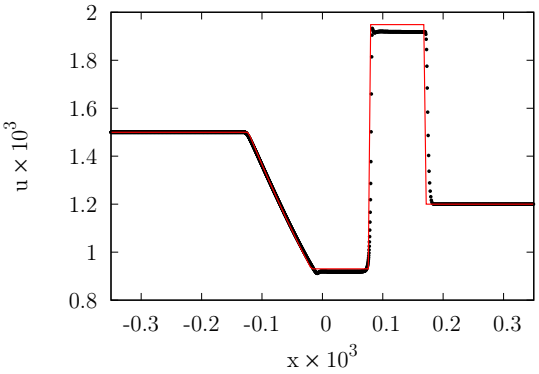

(a) Rest mass density, $\rho_0$



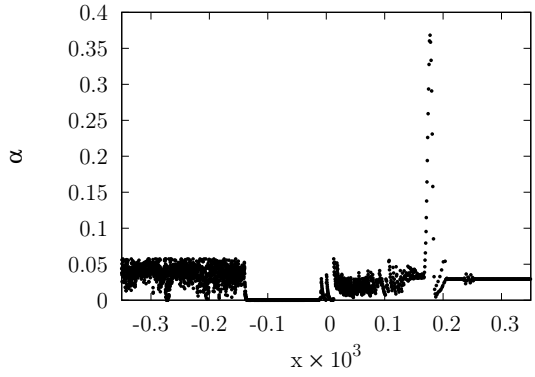

(b) Pressure, $P$




(c) Velocity in the x direction, $v^x$




(d) Specific internal energy, $u$




(e) Dissipation trigger, $\tilde{\alpha}$

Figure 7.1: Results of the Sod shock tube test. The black dots represent the particles of our numerical solution, while the red lines represents the exact solution. We also show the value of our dissipation trigger, $\tilde{\alpha}$

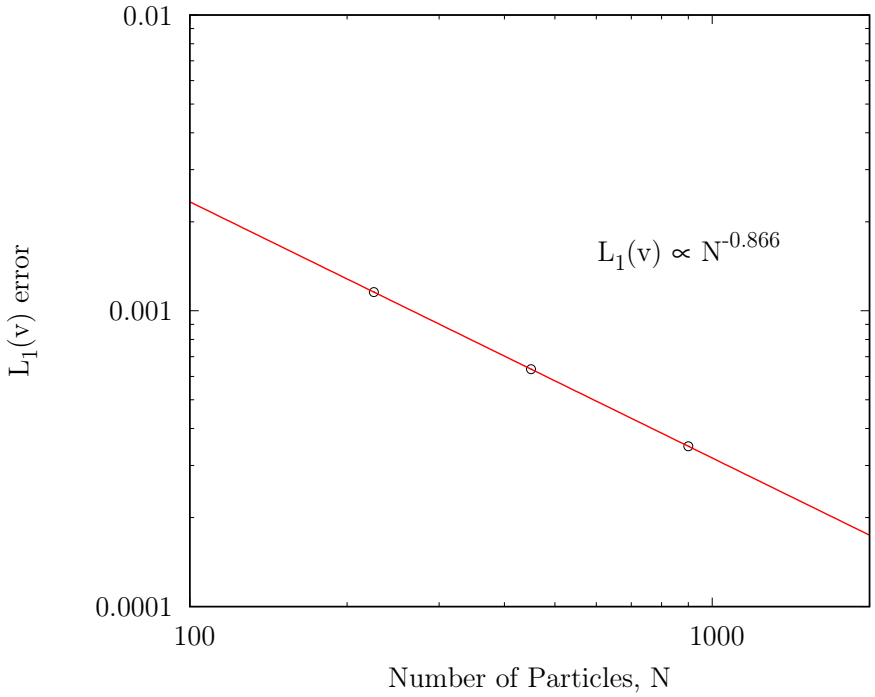



Figure 7.2: Convergence of the nonrelativistic Sod shocktube $L_1(v^x) \propto N^{-.866}$

## 7.2 Two-Dimensional Taylor-Sedov Expansion

The Taylor-Sedov explosion problem serves as another hydrodynamical test of a code's ability to resolve strong shocks. The basis of the problem is the evolution of the spherically symmetric blast wave that results from a delta function perturbation in initial pressure embedded in an otherwise homogeneous medium.

For an initial point-like injection of energy, $E_0$, in a homogeneous background medium with density, $\rho_0$, it has been shown that a radial shock front will move outward from the point with its position as a function of time given by the function,

$$R(t) = C_{\mathcal{D}} \left( \frac{E_0 t^2}{\rho_0} \right)^{\frac{1}{\mathcal{D}+2}}, \tag{7.4}$$

where $C_{\mathcal{D}}$ is a constant of order unity and $\mathcal{D}$ is the number of dimensions of the problem [71, 11]. From this, we see that the shock will propagate with a velocity,

$$v_s(t) = \frac{\mathrm{d}R(t)}{\mathrm{d}t} = \frac{2}{5}\frac{R(t)}{t}. \tag{7.5}$$

At the location of this shock front, mass will pile up forming a dense shell. Through the use of dimensional analysis, Sedov and Taylor then find the form of the fluid variables in the region behind the shock in terms of the similarity variable, $\xi = r/R(t)$ [71, 11]. At the position of the shock, $\xi = 1$, the Rankine-Hugonoit conditions imply that our fluid variables will take the form:

$$\rho(1) = \frac{\Gamma+1}{\Gamma-1}\rho_0 \tag{7.6}$$

$$P(1) = \frac{2}{\Gamma+1}\rho_0 v_s^2 \tag{7.7}$$

$$v_r(1) = \frac{2}{\Gamma+1} v_s \tag{7.8}$$

It can then be shown that the solution in the region behind that shock wave will then take the form,

$$\rho(\xi)/\rho(1) \propto \xi^{\mathcal{D}/(\Gamma-1)} \tag{7.9}$$

$$P(\xi)/P(1) \propto \text{const} \tag{7.10}$$

$$v_r(\xi)/v(1) \propto \xi \tag{7.11}$$

### 7.2.1 Setup

In order to initialize our system for this test, we will deposit an amount of energy, $E_0$, into a small region of radius $\delta r$. The energy will manifest itself as a perturbation to pressure of the form,

$$P_0 = \frac{3(\Gamma - 1)E_0}{(\mathcal{D} + 1)\pi\delta r^{\mathcal{D}}}. \tag{7.12}$$

Our system will have a constant density $\rho_0 = 1$ with an adiabatic index $\Gamma = 5/3$ over the domain $[-1, 1] \times [-1, 1]$ and a constant background pressure $P = 0.001$. In the central region with radius $\delta r$, we will set the pressure to the value $P_0$. For this test, we have chosen $E_0 = .1$ and $\delta r = 0.01$.

The radial density profile at time $t = 2$ is shown in Figure 7.3. We see good agreement with the analytical solution. The density profile shows a slight error in the region where the initially high energy particles are in contact with the initially background region. This discontinuity's radial distance has expanded outward from $r = 0.01$ to $r \approx 0.5$, where the exponential growth in the density levels off slightly, owing to the initial conditions not being a true delta function. Additionally, Figure 7.4 shows the value of our dissipation parameter, $\tilde{\alpha}$, at the same time. We see that, as expected, this parameter is large is in presence of the compression leading the shock front and mild everywhere else in the domain.

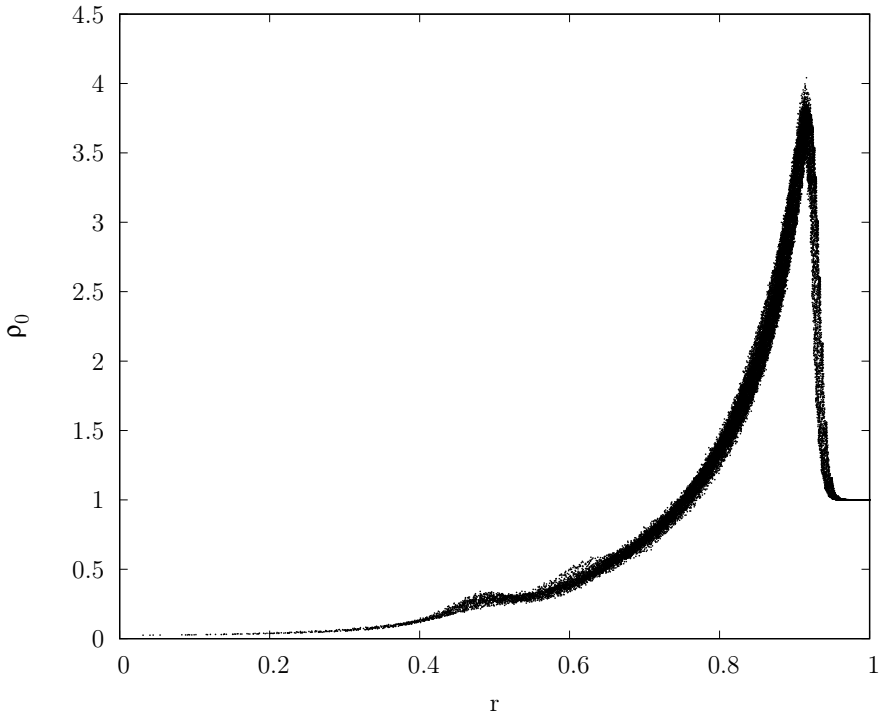



Figure 7.3: Density as a function of radial distance for a Sedov-Taylor blast wave at $t = 2$. The black dots represent the particles of our simulation.

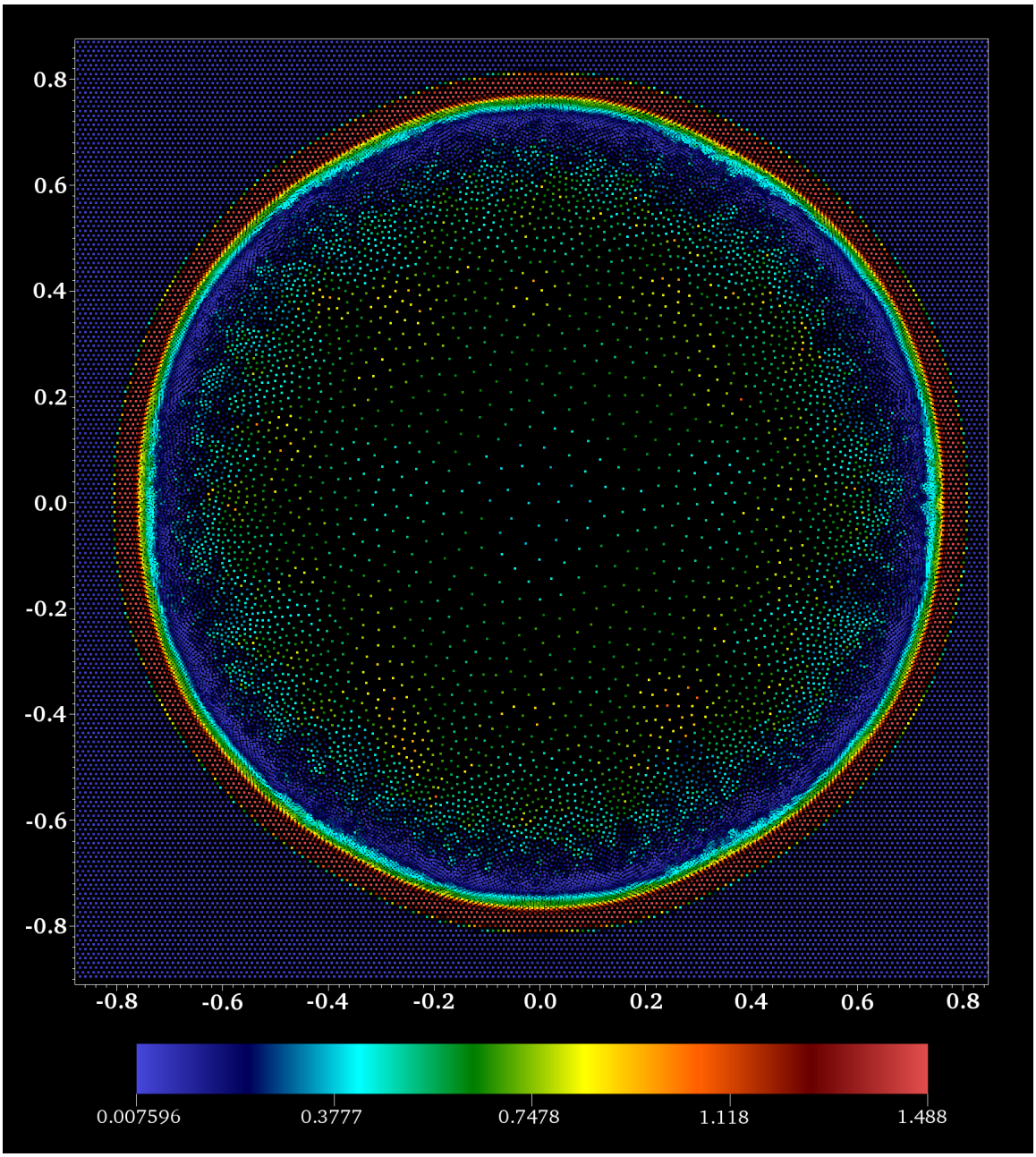



Figure 7.4: Artificial dissipation trigger, $\tilde{\alpha}$, during the propagation of the Sedov-Taylor blast wave.

## *7.3 Two-Dimensional Kelvin-Helmholtz*

SPH methods have been frequently criticized for their inability to properly resolve fluid instabilities due to spurious surface tension forces at fluid boundaries [73, 3, 69, 56]. One such scenario where this is evident is the two dimensional Kelvin-Helmholtz flow. As two fluids of different density experience a shearing flow, the Kelvin-Helmholtz instability is triggered resulting in the turbulent mixing of the fluid [32, 75]. This mixing is the result of the exponential growth of sinusoidal perturbations at the interface. Perturbations of all wavelengths will experience this growth, with the growth rate being proportional to the wavenumber [32]. Such perturbations may be either explicitly seeded in the initial conditions or allowed to develop from numerical noise [53].

### 7.3.1 Setup

For this test, we will initialize the fluid in two strips on the periodic domain $[-1, 1] \times [-1, 1]$. For the region with $|y| < 0.3$, the fluid has a density $D^\star = 1$ and is given a velocity in the positive $x$-direction, $v^x = 0.2$. The fluid outside this region is then initialized such that the density is half that of the inner strip, $D^\star = 0.5$, and it has a velocity equal and opposite $v^x = -0.2$. This sets up a shearing interface between the two fluid regions at $y = \pm.3$. A uniform initial pressure, $P_0 = 10$, is used everywhere in the domain and an adiabatic index $\Gamma = 5/3$ is used. In future tests, it would be preferable to smooth the initial discontinuities in density and velocity over a small region around the interface. This has been shown to produce more consistent results, allowing for easier comparison with other codes [57].

In our first test, we will seed the instability with a velocity perturbation of the form

$$v^y(x) = v_{y,0} \sin(2\pi x), \tag{7.13}$$

with $v_{y,0}$ taking a value of 0.01. Such a perturbation is enough to explicitly trigger

the growth of the lowest mode of the instability. The results of this test, at $t = 4$, are shown in Figure 7.5. In this test, we see the healthy growth of the instability, resulting in the turbulent mixing of the two regions of the fluid. While the fluid still experiences some spurious surface tension, this test has yielded much better results than those of more traditional SPH methods [53].

We will now remove the initial velocity perturbation (e.g $v_{y,0} = 0$) and allow the growth of the instability from purely numerical noise. This result, at $t = 2$, is shown in Figure 7.6. In this case, we see the growth of a higher mode of the instability than in the seeded case.

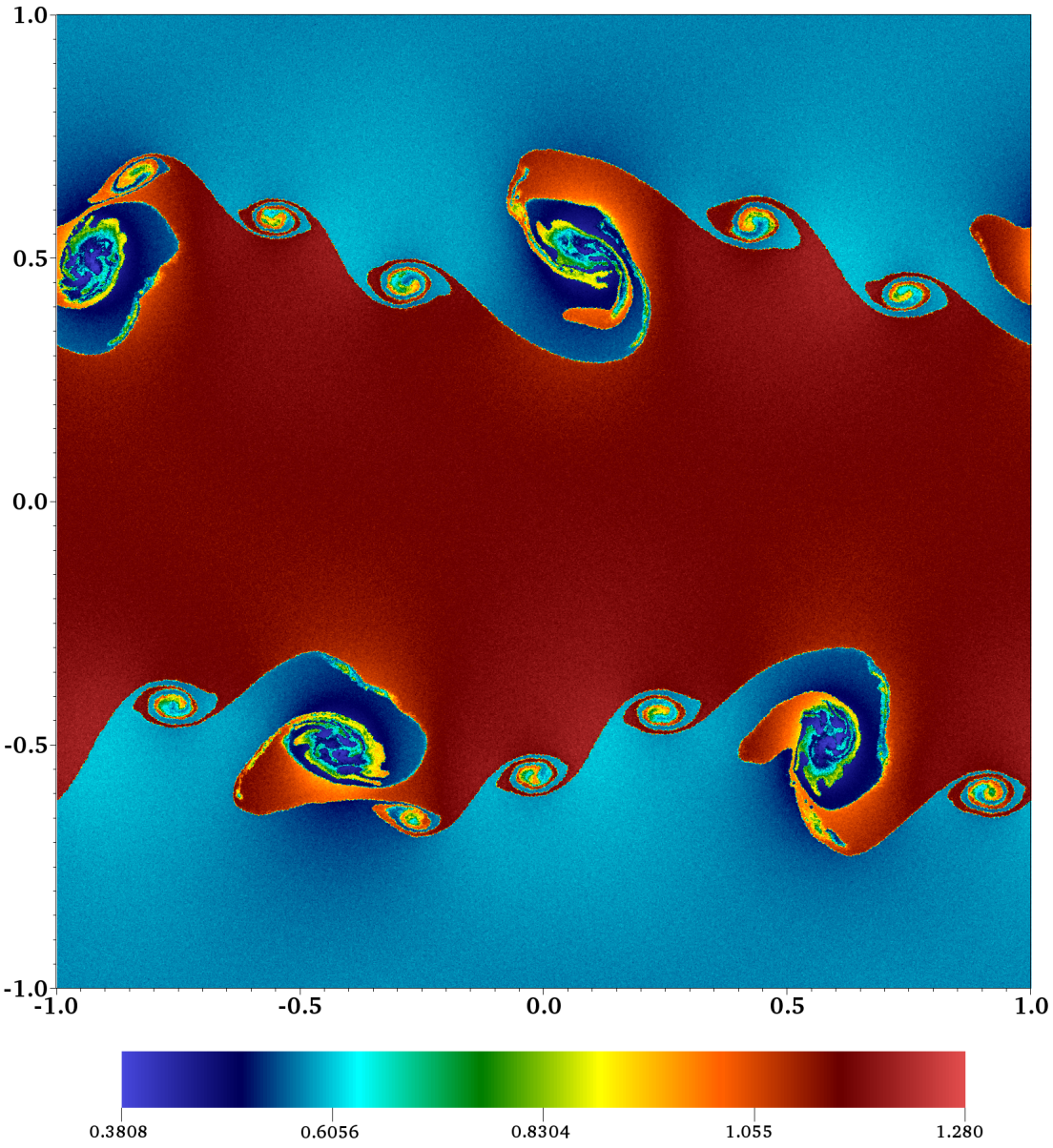



Figure 7.5: Kelvin-Helmholtz simulation at $t = 4$. The initial conditions were perturbed using Equation 7.13 with $v_{y,0} = 0.01$

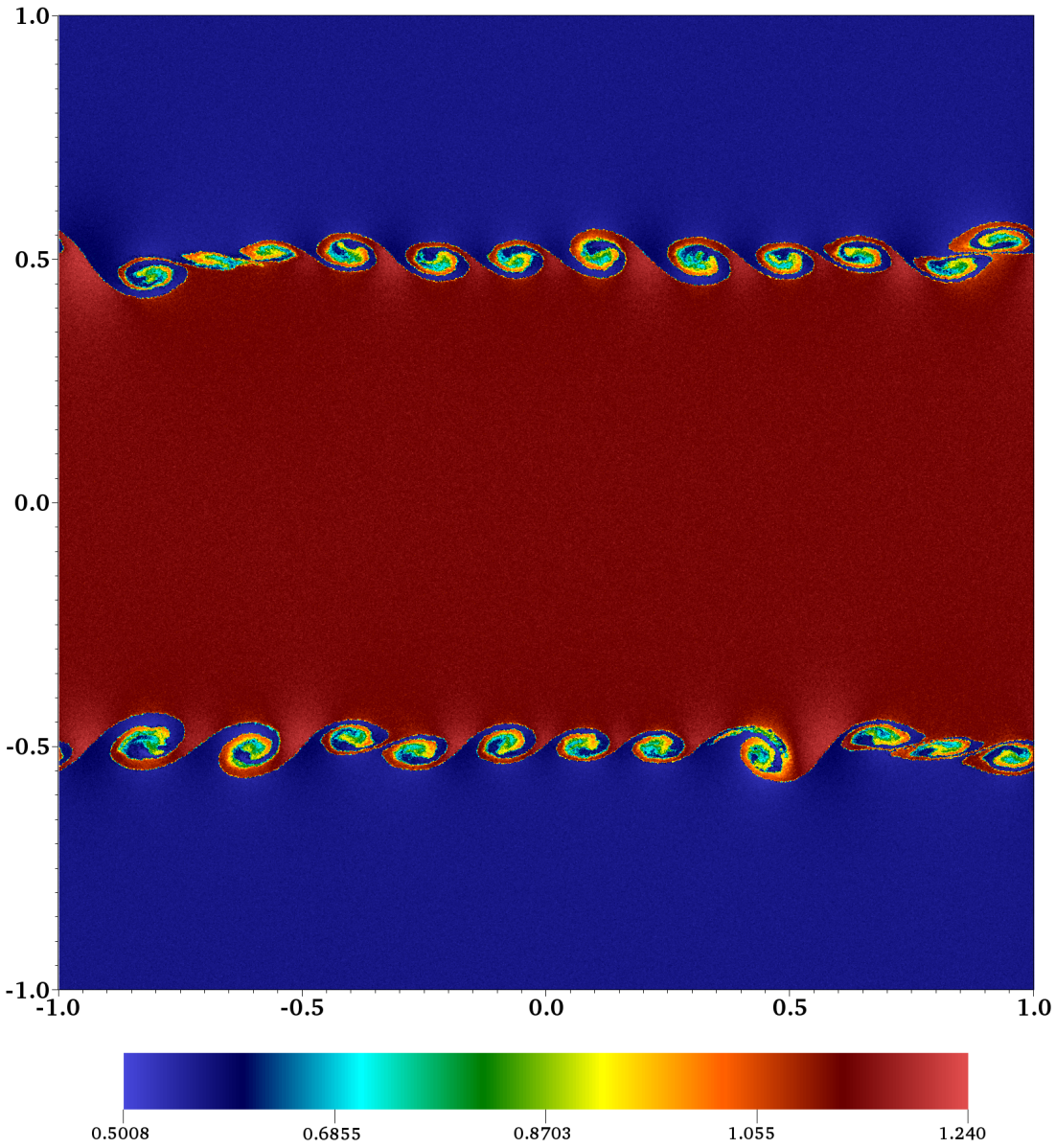



Figure 7.6: Kelvin-Helmholtz simulation at $t = 2$. The initial conditions were unperturbed and the instability was allowed to grow from numerical noise.

## 7.4 Two-Dimensional Gresho-Chan Vortex

We will now turn our attention from dynamical tests to a test that is a steady state solution, the Gresho-Chan vortex test [30]. This test is considered a difficult one for hydrodynamics codes, particularly SPH codes, as the traditional formalisms fail to converge to the correct solution [69]. In this test, a stationary vortex is evolved from a stable equilibrium configuration. The vortex is initialized such that forces arising from the fluid's pressure gradients are matched with those of its centrifugal force. As a result, any deviations from the initial state are numerical artifacts. While the vortex should be stable for many rotations, in practice it will eventually decay due to numerical viscosity. This makes the Gresho-Chan vortex a useful test of our artificial viscosity. Moreover, traditional SPH schemes perform very poorly on this test, slowly converging to the wrong solution [69].

### 7.4.1 Setup

For this test, we must define a characteristic radius, $R_0$. We then initialize the fluid such that the azimuthal component of the fluid's velocity rises linearly and reaches a maximum value of $v_0$ at $r = R_0$. From here, the azimuthal velocity of the fluid falls linearly back to zero at $r = 2R_0$, i.e.:

$$v_\phi(r) = v_0 \begin{cases} r_0 & 0 \leq r_0 \leq 1 \\ 2 - r_0 & 1 < r_0 \leq 2 \\ 0 & r_0 > 2 \end{cases} \tag{7.14}$$

where we have defined $r_0 = r/R_0$.

We then balance the forces resulting from the pressure gradient with the centrifugal force, resulting in the the following form of the pressure in the Newtonian

regime,

$$P(r) = P_0 + \begin{cases} \frac{1}{2}v_0^2 r_0^2 & 0 \leq r_0 \leq 1 \\ 4v_0^2(\frac{r_0^2}{8} - r_0 + \ln r_0 + 1) & 1 < r_0 \leq 2 \\ 4v_0^2\left(\ln 2 - \frac{1}{2}\right) & r_0 > 2 \end{cases} \tag{7.15}$$

Non-relativistic hydrodynamics simulations traditionally use the parameters $\{v_0, R_0\} = \{1, 0.2\}$ but such a velocity will clearly not work for our relativistic code where $c = 1$. To approach the Newtonian limit, we rescale these parameters (similar to Section 7.1.1) by a factor of $10^{-3}$ [62], i.e. $\{v_0, R_0\} = \{10^{-3}, 2 \times 10^{-4}\}$. Additionally, our fluid is chosen to have an adiabatic index $\Gamma = 5/3$ and a constant density such that $\rho_0 = 1$ everywhere. For this test, we will use periodic boundary conditions on a domain of $[-0.0005, 0.0005] \times [-0.0005, 0.0005]$.

Figure 7.7 shows the magnitude of the velocity at the time $t = 1$ relative to the steady state solution. We see that our method has not been excessively dissipative as the velocity peak at $r = 0.0002$ is still relatively well resolved. The resolution of this peak is very good relative to traditional SPH methods (as shown in [69]) due to both the conservative form of our SPH equations (as derived from the Lagrangian) and our method of artificial viscosity triggering.

### 7.4.2 Convergence

In the case of the Gresho-Chan vortex, we will test our convergence by once again using the $L_1$ norm of the velocity, Equation 7.3. In doing this, we will follow the method of [69] and radially bin the velocities of the particles into $N/2$ linearly spaced bins. We will check the average velocity of each bin against the steady state solution at the center of each bin, $r_b$, as

$$L_1(|v|) = \frac{1}{v_0 N_{\text{bins}}} \sum_b^{N_{\text{bins}}} |\bar{v}_b - v(r_b)|, \tag{7.16}$$

where $\bar{v}$ is the average velocity of all particles in bin $b$, and $v(r_b)$ is the exact solution, Equation 7.14, at the center of the bin. We will test this measure of convergence

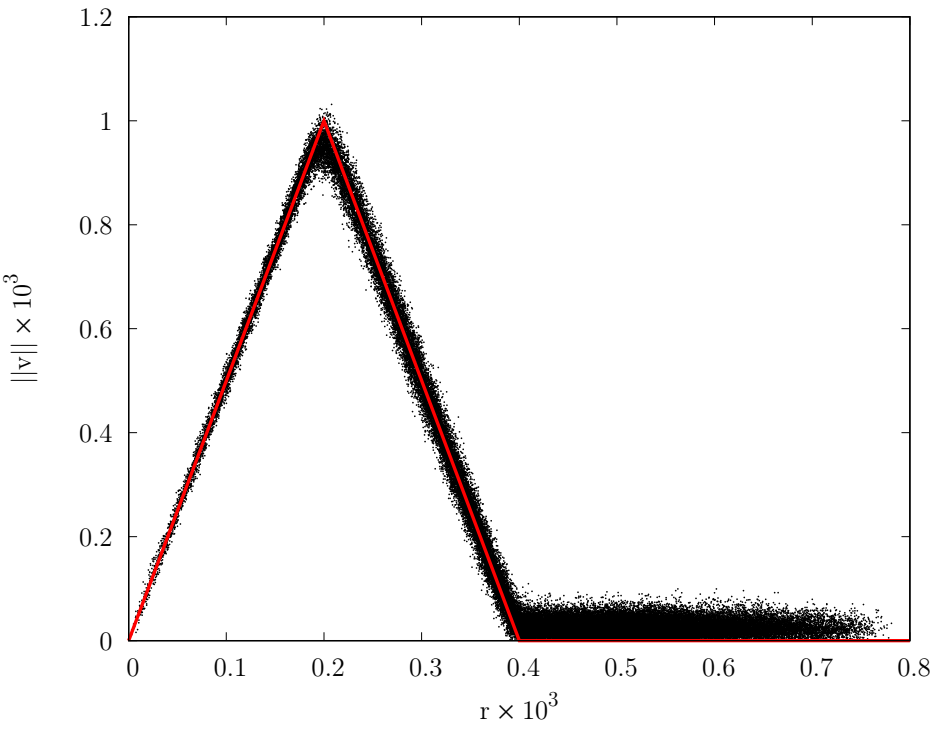



Figure 7.7: Magnitude of the velocity as a function of radius at $t = 1$. The black dots represent our SPH particles while the red line represents the steady state solution, Equation 7.14.

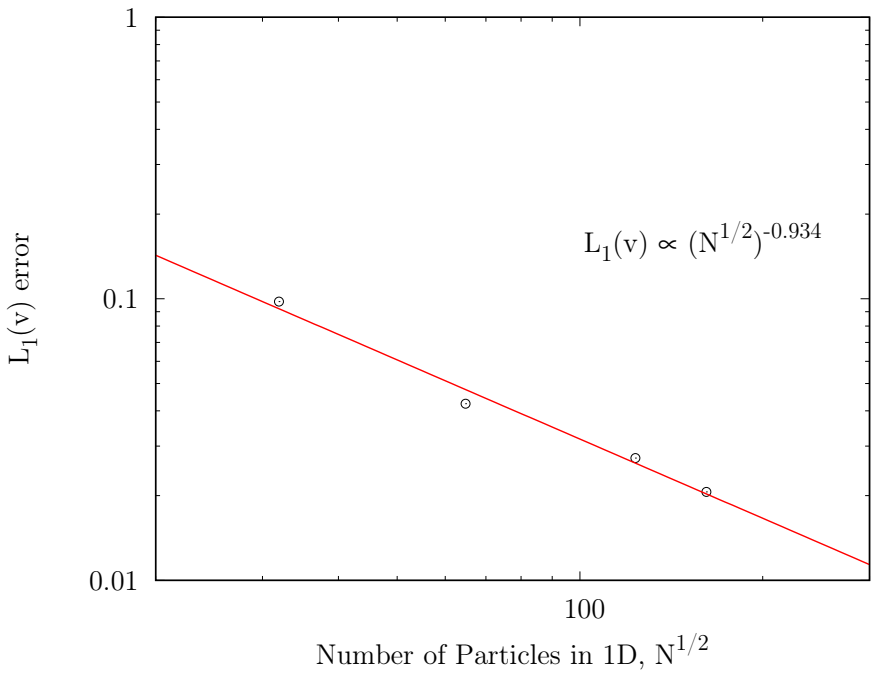



Figure 7.8: Convergence of the Gresho-Chan vortex, $L_1(v^x) \propto \sqrt{N}^{-.936}$

relative to the effective number of particles that span one dimension, $\sqrt{N}$. Figure 7.8 shows the results of this convergence test. Unlike traditional SPH formalisms, our code converges to the correct solution. Also, due to our lack of excessive dissipation, we see a rate of convergence that is quicker than methods that apply dissipation everywhere [62].

## *7.5 One-Dimensional Relativistic Shock Tube*

Having shown our code produces correct results in various test at Newtonian velocities, we will turn our attention to a common problem of relativistic hydrodynamics. Similar to the Newtonian Sod shock tube (Section 7.1), we will test our code's agreement with the exact solution for shocks having velocities in the special relativistic regime. In this regime, we may obtain exact solutions using the method of Centrella and Wilson [17]. In the general case, the initial discontinuity gives rise to a relativistic blast wave and a dense shell of matter. This dense shell moves at relativistic speeds

and has a constant density between the shock front and contact discontinuity.

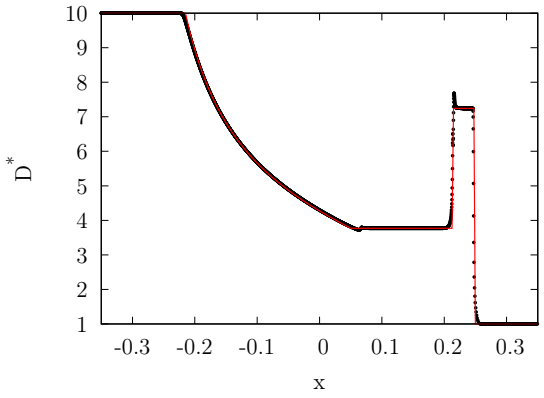
### 7.5.1 Setup

The standard relativistic shock tube is a commonly used benchmark for relativistic codes and exhibits a maximum Lorentz factor of $W_{\max} = 1.4$ [74]. For a fluid with an adiabatic index $\Gamma = 5/3$, this test consists of two fluid domains. In the standard case, the left domain is initially in the state $\{P, v_x, D\}_L = \{40/3, 0, 10\}$, and the right domain is initially in the state $\{P, v_x, D\}_L = \{10^{-6}, 0, 1\}$. For this test, we will once again smooth the initial discontinuity over the ten particles closest to the interface using Equation 7.1.

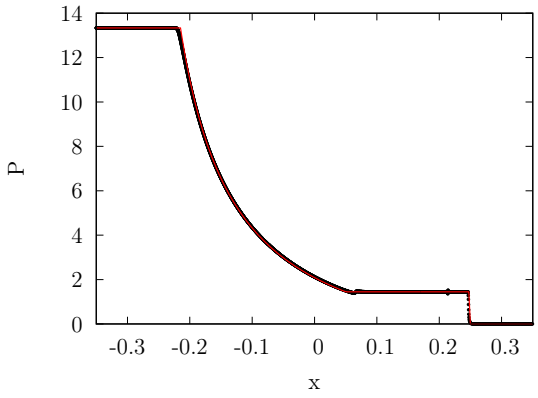
The results of this test using $N = 5500$ particles are shown in Figure 7.9. Our results show excellent agreement with the analytical solution. As was the case in the non-relativistic Sod shock tube, we see that our dissipation trigger, $\tilde{\alpha}$, reaches its maximum value at the shock front and is otherwise only mildly triggered in the presence of velocity noise.

### 7.5.2 Convergence

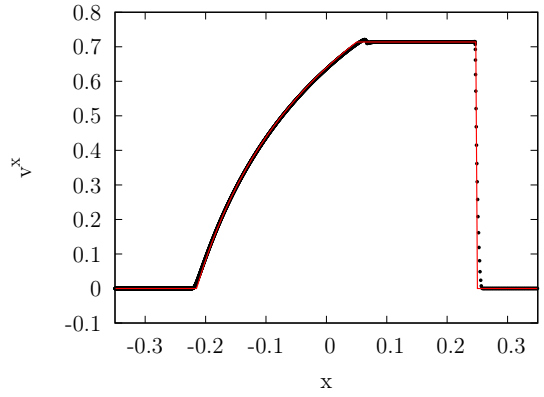
For this test, the convergence of the $L_1(v^x)$ error (Equation 7.3) as a function of the number of particles in our simulation domain is shown in Figure 7.2. In this test, our code shows much better convergence than the nonrelativistic case. Surprisingly, we even see a convergence rate that is slightly better than the optimal convergence rate of $N^{-1}$. This is likely caused by the fact that we have smoothed our initial discontinuity over a constant number of particles at the interface. As a result of this, we are improving the quality of our initial conditions as the resolution increases.



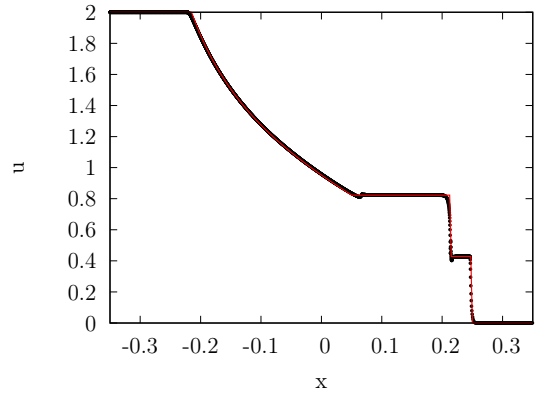

(a) Relativistic density, $D^\star$



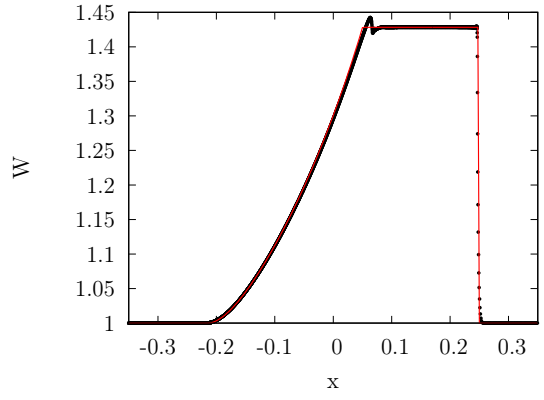

(b) Pressure, $P$



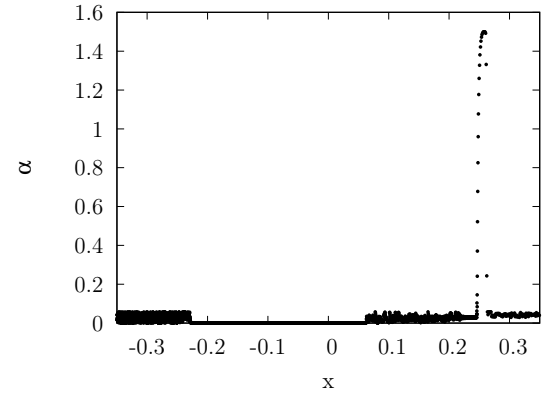

(c) Coordinate velocity in the x-direction, $v^x$




(d) Specific internal energy, $u$




(e) Lorentz Factor, $W$




(f) Dissipation trigger, $\alpha$

Figure 7.9: Results of the Relativistic shock tube test. The black dots represent the particles of our numerical solution, while the red lines represents the exact solution of [17]. We also show the value of our dissipation trigger, $\tilde{\alpha}$.

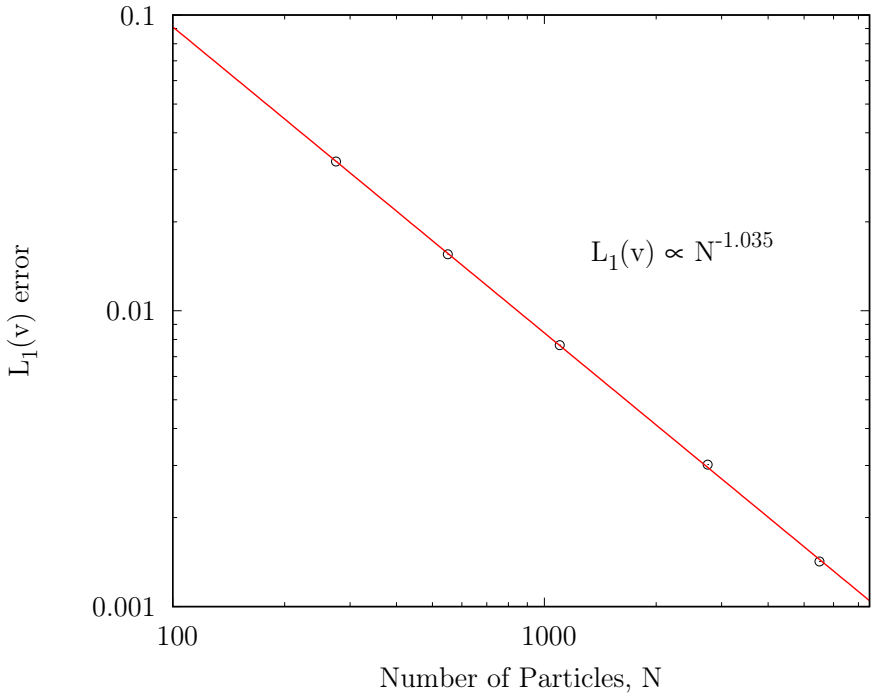



Figure 7.10: Convergence of the relativistic sod shock tube $L_1(v^x) \propto N^{-1.035}$

## 7.6 Bondi Accretion onto a Schwarzschild Black Hole

Now we will test our code in a non-flat spacetime. One classical test in this case is that of Bondi accretion onto a Schwarzschild black hole. A model of the simplest accretion scenario, Bondi accretion describes the steady accretion flow onto a Newtonian point mass of a gas that, at infinity, has a constant density and is at rest [12]. In this situation, it can be shown that the rate of accretion onto the central object takes the form,

$$\dot{M} \approx \pi R^2 \rho_\infty c_s, \tag{7.17}$$

where $R$ is the effective radius of the accreting object. By equating the object's escape velocity to the speed of sound, one can estimate the radius $R$ as

$$R \approx \frac{2M}{c_s^2}, \tag{7.18}$$

yielding the accretion rate:

$$\dot{M} = \frac{4\pi \rho_\infty M^2}{c_s^3}. \tag{7.19}$$

A generalization of this problem in a Schwarzschild spacetime with mass $M$ was first detailed analytically by Michel [44]. Using the gamma-law EoS (Equation 3.17) we define

$$T = P/\rho = (\Gamma - 1)u \tag{7.20}$$

for convenience. We then begin by assuming spherical symmetry to write the conservation of baryon number (Equation 3.15) as

$$T^n U^r r^2 = C_1, \tag{7.21}$$

where $n$ is the polytropic index ($n = 1/(\Gamma - 1)$). This assumption will also allow us to write the equation for conservation of mass energy (Equation 3.16) as

$$[1 + (1 + n)T]^2 [1 - 2M/r + (U^r)^2] = C_2. \tag{7.22}$$

By providing the value of the constants $C_1$ and $C_2$, Equations 7.21 and 7.22 will allow us to calculate the values of $T$ and $U^r$ everywhere. In order to obtain the constants $C_1$ and $C_2$, we begin by differentiating Equations 7.21 and 7.22,

$$\frac{\mathrm{d}U^r}{U^r}\left[c_s^2 - \frac{(U^r)^2}{1-2M/r+(U^r)^2}\right] + \frac{\mathrm{d}r}{r}\left[2c_s^2 - \frac{M}{r(1-2M/r+(U^r)^2)}\right] = 0. \tag{7.23}$$

We will then define a critical point, $r_c$, at which the flow changes from subsonic to supersonic (i.e. $\mathcal{M} = 1$). At this critical point, the following conditions hold:

$$(U_c^r)^2 = \frac{M}{2r_c}, \tag{7.24}$$

$$c_{s,c}^2 = \frac{(U_c^r)^2}{1-3(U_c^r)^2}. \tag{7.25}$$

Now, having choosing a critical radius, $r_c$, and using the relation,

$$\frac{(1+n)T_c}{n[1+(1+n)T_c]} = c_{s,c}^2 = \frac{(U_c^r)^2}{1-3(U_c^r)^2}, \tag{7.26}$$

we may pin the values for the constants $C_1$ and $C_2$. Having determined the constants $C_1$ and $C_2$, we can then uniquely define the radial functions $U^r(r)$ and $T(r)$ everywhere by using the relations:

$$U^r = \frac{C_1}{T^n r^2}, \tag{7.27}$$

$$C_2 = [1+(1+n)T]^2\left(1 - \frac{2M}{r} + \frac{C_1^2}{r^4 T^{2n}}\right). \tag{7.28}$$

We must now determine the value of our primitive variables using the fact that

$$\frac{\rho}{\rho_\infty} = \left(\frac{T}{T_\infty}\right)^n \tag{7.29}$$

or equivalently,

$$\rho = (\rho_\infty / T_\infty^n) T^n = K T^n \tag{7.30}$$

where we have introduced the constant $K$. $K$ is an indicator of the particular adiabat of our fluid. With this, we now have knowledge of $\rho_0$, $P = T/\rho$, and $U^r$ over the whole radial domain, which fully describes our fluid system when coupled with our equation of state.

### 7.6.1 Setup

For this test, we initialize our fluid in the domain $r \in [2.01, 9]$ having chosen a critical radius, $r_c = 5$, and the $K = 1$ adiabat. The fluid is chosen to have adiabatic index $\Gamma = 1.4$. The density distribution of the Bondi solution is tabulated into a radial cumulative distribution function, allowing us to probabilistically place particles of equal mass radially with the correct density profile. We then evolve the fluid in time to check that the critical radius remains stable at its location [35].

The results at $t = 1$ are shown in Figure 7.11. After undergoing this evolution, we will check for the stability of the critical radius $r_c$. For this, we use polynomial interpolation to define the value of the function $f(r) = \mathcal{M} - 1$ everywhere. Using a standard root finding procedure on $f(r)$ will allow us to locate the position where the flow has crossed from subsonic to supersonic (i.e. $f(r) = 0$). In this simulation, that location was found at $r = 4.976$, representing an error of only 0.5% when compared to the initial location over the course of 100,000 iterations.

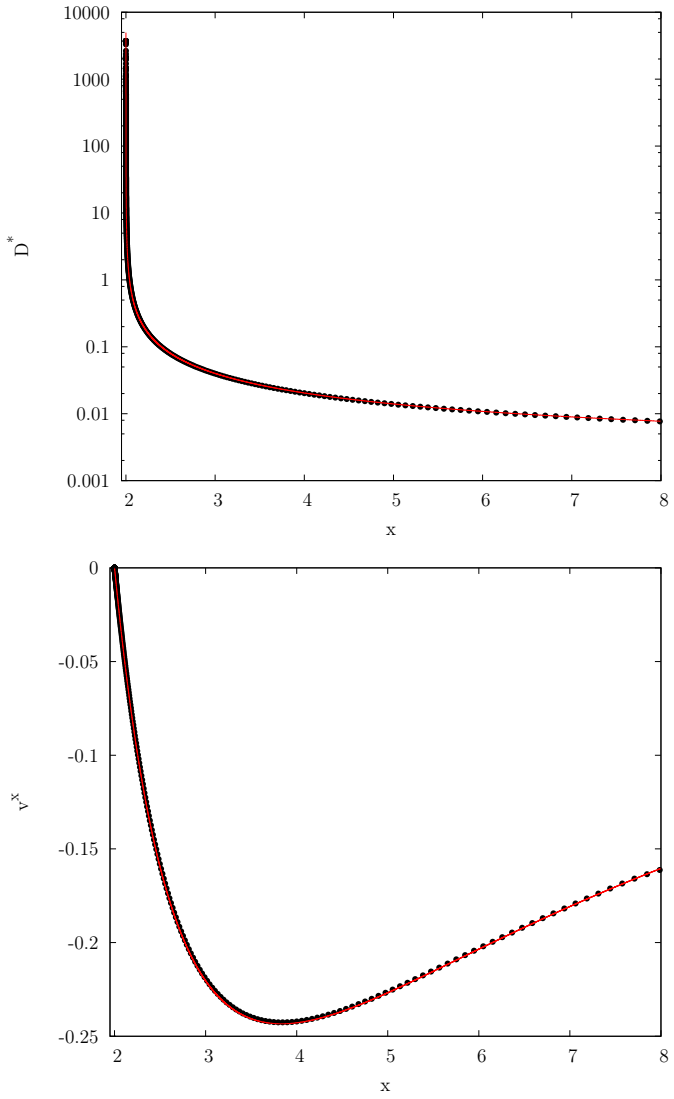



Figure 7.11: Bondi accretion onto a Schwarzschild black hole: The top figure shows the density variable, $D^\star$ and the bottom figures shows the radial velocity $v^r$ after evolution until $t = 1$. The black dots represent the SPH particles while the red line shows the exact solution.

# CHAPTER VIII

# CONCLUSIONS AND FUTURE WORK

In this work, we have derived a set of general relativistic hydrodynamics equations suitable for a Lagrangian treatment of a fluid. By using the smoothed particle hydrodynamics method, we are able to simulate such a fluid in the presence of strong gravity. Moreover, this method of solving the equations of general relativistic hydrodynamics has proven to be well suited for modern supercomputers due to its highly parallel nature. We have tested the robustness of our code in various standard Newtonian and relativistic tests. In its current state, our new code gives us the capacity to model various systems of interest in numerical astrophysics, but holds a much greater potential in its ability to interface with a dynamical spacetime solver (e.g the Einstein Toolkit [4, 41] or the Georgia Tech Maya code [33, 34]).

In chapter 3, we have derived evolution equations of Lagrangian hydrodynamics in the 3+1 decomposition used by numerical relativity. These equations are consistent with those of previous Lagrangian methods but have additionally been formulated to fit naturally within the framework of a 3+1 spacetime evolution code. We then reviewed many of the modern methods being used in the SPH community in Chapter 4. Having derived the basics of the SPH approximation, we have shown a more accurate means of calculating derivatives in the SPH formalism, based on an integral approximation for derivatives.

With all of our tools in hand, Chapter 5 presented the final form of our equations. By means of a variational derivation, we discretized our continuum equations of general relativistic hydrodynamics. From this point, we added in a prescription for artificial viscous pressure that can be added consistently to both the hydrodynamical

and gravitational terms of the evolution equations. In order to improve our numerical convergence, we outlined a method by which this artificial viscosity is triggered only in the presence of a shock front or unphysical noise in the velocity field.

Then, in Chapter 6, we described the computational methods that were needed in order to evolve such a particle system. With the nearest neighbor searching algorithm being the most computationally complex step in any SPH code, we outlined a hash table based method that has proven itself not only fast, but highly parallel. We outlined the ORB method which allowed us to decompose the physical domain over the many processors on a distributed system while dynamically load balancing each processes load using timing statistics.

Finally, in Chapter 7, we have shown the validity of our formalism and code by applying it to many standard problems in numerical hydrodynamics. Our code has performed well in the case of various problems involving shocks, including both the standard and special relativistic shock tube problems. We have also shown good results in the particularly challenging Gresho-Chan vortex test, in which accurate control of dissipation is required to maintain the sharp velocity peak of the steady state solution. We were also able to apply our code to the standard test case of seeded Kelvin-Helmholtz instabilities. In this test, we were able to simulate the growth of the Kelvin-Helmholtz billows with fairly good accuracy for an SPH method. Finally, we have shown that our code is able to recreate the steady state solution of Bondi accretion onto a black hole. In this non-flat spacetime, we were able to test our code's ability to handle of the terms in our formalism resulting from the curvature of spacetime.

### 8.0.1 Future Work

With our general relativistic SPH method formulated and tested, the next step is to fully couple it with a spacetime evolution code. Since the equations were formulated

using the 3+1 decomposition of the metric, we can cleanly read in the interpolated values of the metric as computed by such a code and let the fluid "ride" on the dynamical spacetime as a tracer. More interestingly, we would like to fully couple the Einstein equations with the equations of hydrodynamics by providing the spacetime evolution code with its matter source terms, calculated based on the SPH fluid. To do this, one could naively evaluate the SPH approximation at the locations where the spacetime is defined,

$$T_{\mu\nu}(\vec{r}_a) = \sum_b \frac{m_b}{D_b^\star}(\rho_0 \omega U_\mu U_\nu + P g_{\mu\nu})_b W_{ab}(h_b), \tag{8.1}$$

but it is likely that this will lead to considerable noise and conflict between the inconsistent resolution of the particles and Eulerian grid. Further work is needed to develop a proper method to deposit the mass and energy onto an Eulerian grid in order to calculate the source terms. Once a method for this has been prescribed, our SPH code could easily be used in problems where the fluid's self-gravity is not negligible. Such problems include NSNS binaries, NSBH binaries, white dwarf mergers, and even the tidal disruption of stars by black holes.

# APPENDIX A

# ANCILLARY MATERIAL

## *A.1 General Relativistic SPH From a Lagrangian*

In this section, we will detail the derivation of the general relativistic SPH equations with variable smoothing length using only the Lagrangian of a relativistic fluid and our prescription for calculating the density in the SPH approximation. This method was first outlined in [70].

First, having established the SPH discretization of our density variable $D^\star$ (Equation 5.1) we will calculate a few quantities that will prove useful in our variational derivation of the evolution equations. We see that the spatial derivative of $D^\star$, allowing for the smoothing length to vary, takes the form:

$$\begin{aligned}\frac{\partial D_b^\star}{\partial x_a^i} &= \sum_c m_c \left[\frac{\partial W_{bc}(h_b)}{\partial r_{bc}}\frac{\partial r_{bc}}{\partial x_a^i} + \frac{\partial W_{bc}(h_b)}{\partial h_b}\frac{\partial h_b}{\partial x_a^i}\right] \\ &= \sum_c m_c \left[\frac{\partial W_{bc}(h_b)}{\partial r_{bc}}\frac{\partial r_{bc}}{\partial x_a^i} + \frac{\partial W_{bc}(h_b)}{\partial h_b}\frac{\partial h_b}{\partial D_b^\star}\frac{\partial D_b^\star}{\partial x_a^i}\right]. \end{aligned} \tag{A.1}$$

By collecting the terms $\partial D^\star/\partial x_a^i$, we find that

$$\begin{aligned}\frac{\partial D_b^\star}{\partial x_a^i} &= \frac{1}{\Omega_b}\sum_c m_c \frac{\partial W_{bc}(h_b)}{\partial r_{bc}}\frac{\partial r_{bc}}{\partial x_a^i} \\ &= \frac{1}{\Omega_b}\sum_c m_c \frac{\partial W_{bc}(h_b)}{\partial x_b^i}(\delta_{ba} - \delta_{ca}), \end{aligned} \tag{A.2}$$

where we have introduced the variable,

$$\Omega_b = 1 - \frac{\partial h_b}{\partial D_b^\star}\sum_c m_c \frac{\partial W_{bc}(h_b)}{\partial h_b}. \tag{A.3}$$

We may similarly take the time derivative of $D^\star$, noting that the time derivative $\mathrm{d}r_{ab}/\mathrm{d}t = v_{ab}^i(\hat{r}_{ab})_i$ where $v_{ab}^i = v_a^i - v_b^i$:

$$\frac{\mathrm{d}D_b^\star}{\mathrm{d}t} = \frac{1}{\Omega_b}\sum_c m_c v_{bc}^i \frac{\partial W_{bc}(h_b)}{\partial x_b^i}. \tag{A.4}$$

We will also make use of the first law of thermodynamics, which at constant entropy states that,

$$\left.\frac{\partial u}{\partial \rho_0}\right|_s = \frac{P}{\rho_0^2}. \tag{A.5}$$

We may now derive our momentum and energy evolution equation using the variational method first outlined in [70]. We will find the discrete form of the Lagrangian for a general relativistic fluid [25],

$$\begin{aligned} L &= -\int T^{\mu\nu} U_\mu U_\nu \sqrt{-g} dV \\ &= -\int \rho_0 (1+u) \alpha \sqrt{\gamma} dV \\ &= -\int \frac{\alpha(1+u)}{W} D^\star \mathrm{d}V. \end{aligned} \tag{A.6}$$

By approximating our volume element as $\mathrm{d}V \approx \Delta V = m/D^\star$, and applying the same discretization as Equation 4.4, we arrive at the following form for the Lagrangian,

$$L \approx -\sum_b m_b \frac{\alpha_b (1+u_b)}{W_b}. \tag{A.7}$$

With this, the Euler-Lagrange equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\partial L}{\partial v^i}\right) - \frac{\partial L}{\partial x^i} = 0, \tag{A.8}$$

will provide us with a symmetric version of our evolution equations that account for variable smoothing lengths.

### A.1.1 The Momentum Equation

We will begin with the Euler-Lagrange equation for the momentum evolution. To start we will look at the canonical momentum, $p_i$, for a particle $a$:

$$\begin{aligned} p_{i,a} &\equiv \frac{\partial L}{\partial v_a^i} \\ &= -\frac{\partial}{\partial v_a^i} \sum_b m_b \frac{\alpha_b (1+u_b)}{W_b} \\ &= -\sum_b m_b \alpha_b \frac{\partial}{\partial v_a^i} \left( \frac{(1+u_b)}{W_b} \right) \\ &= -\sum_b m_b \alpha_b \left[ \frac{1}{W_b} \frac{\partial u_b}{\partial v_a^i} - \frac{1+u_b}{W_b^2} \frac{\partial W_b}{\partial v_a^i} \right]. \end{aligned} \tag{A.9}$$

We must now break up the derivative, $\partial u / \partial v^i$, and use the first law of thermodynamics (Equation A.5) to show that,

$$\begin{aligned} \frac{\partial u_b}{\partial v_a^i} &= \frac{\partial u_b}{\partial \rho_{0,b}} \frac{\partial \rho_{0,b}}{\partial v_a^i} \\ &= \left( \frac{P}{\rho_0^2} \right)_b \frac{\partial}{\partial v_a^i} \left( \frac{D_b^\star}{\sqrt{\gamma_b} W_b} \right) \\ &= \left( \frac{P D^\star}{\sqrt{\gamma} \rho_0^2} \right)_b \frac{\partial}{\partial v_a^i} \left( \frac{1}{W_b} \right) \\ &= \left( \frac{-P}{\rho_0 W} \right)_b \frac{\partial W_b}{\partial v_a^i}. \end{aligned} \tag{A.10}$$

All that remains is the derivative of the Lorentz factor, $W$, with respect to the velocity, $v^i$:

$$\begin{aligned} \frac{\partial W_b}{\partial v_a^i} &= \frac{\partial}{\partial v_a^i} \left[ \left( 1 - \gamma_{jk} V^j V^k \right)_b^{-1/2} \right] \\ &= \frac{\left( 1 - \gamma_{jk} V^j V^k \right)_b^{-3/2}}{2} \frac{\partial}{\partial v_a^i} \left( \gamma_{jk} V^j V^k \right)_b \\ &= \left( \frac{W^3 \gamma_{jk}}{2} \right)_b \frac{\partial (V^j V^k)_b}{\partial v_a^i} = \left( W^3 \gamma_{jk} V^j \right)_b \frac{\partial V_b^k}{\partial v_a^i} \\ &= \left( \frac{W^3 \gamma_{jk} V^j}{\alpha} \right)_b \frac{\partial v_b^k}{\partial v_a^i} = \left( \frac{W^3 \gamma_{jk} V^j}{\alpha} \right)_b \delta_{ki} \delta_{ab} \\ &= \left( \frac{W^3 V_i}{\alpha} \right)_b \delta_{ab} \end{aligned} \tag{A.11}$$

Now, by plugging Equations A.10 and A.11 into Equation A.9, we see that the canonical momentum takes the form:

$$\begin{aligned} p_{i,a} &= -\sum_b m_b \alpha_b \left[\frac{-P}{\rho_0 W^2} - \frac{1+u}{W^2}\right]_b \frac{\partial W_b}{\partial v_a^i} \\ &= \sum_b \left(\frac{m\alpha\omega}{W^2}\right)_b \left(\frac{W^3 V_i}{\alpha}\right)_b \delta_{ab} \\ &= (m\omega W V_i)_a. \end{aligned} \tag{A.12}$$

It is now apparent that our momentum variable, $S_i$ is related to $p_i$ as $p_i = mS_i$. Since $m$ is constant in time we can rewrite the Euler-Lagrange equations (Equation A.8) in terms of $S_i$ as

$$\begin{aligned} \frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} &= -\frac{1}{m_a}\frac{\partial L}{\partial x_a^i} \\ &= -\frac{1}{m_a}\sum_b m_b \frac{\partial}{\partial x_a^i}\left(\frac{\alpha(1+u)}{W}\right)_b \\ &= -\frac{1}{m_a}\sum_b m_b \left[\left(\frac{1+u}{W}\right)_b \frac{\partial \alpha_b}{\partial x_a^i}\delta_{ab} + \left(\frac{\alpha}{W}\right)_b \frac{\partial u_b}{\partial x_a^i} - \left(\frac{\alpha(1+u)}{W^2}\right)_b \frac{\partial W_b}{\partial x_a^i}\right] \end{aligned} \tag{A.13}$$

We will further simplify Equation A.13 by once again applying the first law of thermodynamics (Equation A.5) to rewrite $\partial u/\partial x^i$,

$$\begin{aligned} \frac{\partial u_b}{\partial x_a^i} &= \frac{\partial u_b}{\partial \rho_{0,b}}\frac{\partial \rho_{0,b}}{\partial x_a^i} = \left(\frac{P}{\rho_0^2}\right)_b \frac{\partial}{\partial x_a^i}\left(\frac{D^\star}{\sqrt{\gamma}W}\right)_b \\ &= \left(\frac{P}{\rho_0^2}\right)_b \left[\left(\frac{1}{\sqrt{\gamma}W}\right)_b \frac{\partial D_b^\star}{\partial x_a^i} - \left(\frac{D^\star}{\sqrt{\gamma}W^2}\right)_b \frac{\partial W_b}{\partial x_a^i} - \left(\frac{D^\star}{2\gamma^{3/2}W}\right)_b \frac{\partial \gamma_b}{\partial x_a^i}\right] \\ &= \left(\frac{P}{\sqrt{\gamma}\rho_0^2 W}\right)_b \left[\frac{\partial D_b^\star}{\partial x_a^i} - \left(\frac{D^\star}{W}\right)_b \frac{\partial W_b}{\partial x_a^i} - \left(\frac{D^\star \gamma^{jk}}{2}\right)_b \frac{\partial \gamma_{jk,b}}{\partial x_a^i}\delta_{ab}\right] \\ &= \left(\frac{P}{\rho_0}\right)_b \left[\frac{1}{D_b^\star}\frac{\partial D_b^\star}{\partial x_a^i} - \frac{1}{W_b}\frac{\partial W_b}{\partial x_a^i} - \frac{\gamma_b^{jk}}{2}\frac{\partial \gamma_{jk,b}}{\partial x_a^i}\delta_{ab}\right], \end{aligned} \tag{A.14}$$

where we have used the identity [8],

$$\frac{\partial \gamma_b}{\partial x_a^i} = (\gamma\gamma^{jk})_b \frac{\partial \gamma_{jk,b}}{\partial x_a^i}\delta_{ab}. \tag{A.15}$$

The spatial derivative of the Lorentz factor is,

$$\begin{aligned}
\frac{\partial W_b}{\partial x_a^i} &= \frac{\partial}{\partial x_a^i}\left[(1-\gamma_{jk}V^jV^k)_b^{-1/2}\right] \\
&= \frac{(1-\gamma_{jk}V^jV^k)_b^{-3/2}}{2}\frac{\partial}{\partial x_a^i}\left(\gamma_{jk}V^jV^k\right)_b \\
&= \frac{W_b^3}{2}\left[V_b^jV_b^k\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\delta_{ab} + (2\gamma_{jk}V^j)_b\frac{\partial V_b^k}{\partial x_a^i}\right] \\
&= W_b^3\left[\frac{V_b^jV_b^k}{2}\frac{\partial\gamma_{jk,b}}{\partial x_a^i} - \left(\frac{V_kV^k}{\alpha}\right)_b\frac{\partial\alpha_b}{\partial x_a^i} + \left(\frac{V_j}{\alpha}\right)_b\frac{\partial\beta_b^j}{\partial x_a^i}\right]\delta_{ab}, \qquad \text{(A.16)}
\end{aligned}$$

where we have used Equation 3.7. With Equations A.14 and A.16 we may write Equation A.13 as follows:

$$\begin{aligned}
\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} &= -\frac{1}{m_a}\sum_b m_b\left[\left(\frac{1+u}{W}\right)_b\frac{\partial\alpha_b}{\partial x_a^i}\delta_{ab}\right. \\
&\left. + \left(\frac{\alpha P}{W\rho_o}\right)_b\left(\frac{1}{D_b^\star}\frac{\partial D_b^\star}{\partial x_a^i} - \frac{1}{W_b}\frac{\partial W_b}{\partial x_a^i} - \frac{\gamma_b^{jk}}{2}\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\delta_{ab}\right) - \left(\frac{\alpha(1+u)}{W^2}\right)_b\frac{\partial W_b}{\partial x_a^i}\right] \\
&= -\frac{1}{m_a}\sum_b m_b\left\{\left(\frac{\alpha\sqrt{\gamma}P}{D^{\star 2}}\right)_b\frac{\partial D_b^\star}{\partial x_a^i} - \left(\frac{\alpha}{W^2}\right)_b\left(1+u+\frac{P}{\rho_0}\right)_b\frac{\partial W_b}{\partial x_a^i}\right. \\
&\left. + \left[\left(\frac{1+u}{W}\right)_b\frac{\partial\alpha_b}{\partial x_a^i} - \left(\frac{\alpha\sqrt{\gamma}P\gamma^{jk}}{2D^\star}\right)_b\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\right]\delta_{ab}\right\} \\
&= -\frac{1}{m_a}\sum_b m_b\left\{\left(\frac{\alpha\sqrt{\gamma}P}{D^{\star 2}}\right)_b\frac{\partial D_b^\star}{\partial x_a^i}\right. \\
&+ (\alpha W\omega)_b\left[\left(\frac{1-1/W^2}{\alpha}\right)_b\frac{\partial\alpha_b}{\partial x_a^i} - \left(\frac{V_j}{\alpha}\right)_b\frac{\partial\beta_b^j}{\partial x_a^i} - \frac{V_b^jV_b^k}{2}\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\right]\delta_{ab} \\
&\left. + \left[\left(\frac{1+u}{W}\right)_b\frac{\partial\alpha_b}{\partial x_a^i} - \left(\frac{\alpha\sqrt{\gamma}P\gamma^{jk}}{2D^\star}\right)_b\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\right]\delta_{ab}\right\}, \qquad \text{(A.17)}
\end{aligned}$$

where we have used the property that $\partial_a W_{ab} = -\partial_b W_{ab}$. At this point it is convenient to break up the equation into two parts such that

$$\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} = \left.\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t}\right|_{hydro} + \left.\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t}\right|_{grav}. \qquad \text{(A.18)}$$

Taking the first term of the summation in Equation A.17,

$$\begin{aligned}\left.\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t}\right|_{hydro} &= -\frac{1}{m_a}\sum_b m_b\left(\frac{\alpha\sqrt{\gamma}P}{D^{\star 2}}\right)_b\frac{\partial D_b^\star}{\partial x_a^i}\\ &= -\frac{1}{m_a}\sum_b m_b\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b\sum_c m_c\frac{\partial W_{bc}(h_b)}{\partial x_b^i}(\delta_{ba}-\delta_{ca})\\ &= -\sum_b m_b\left[\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_a\frac{\partial W_{ab}(h_a)}{\partial x_a^i}+\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b\frac{\partial W_{ab}(h_b)}{\partial x_a^i},\right] \end{aligned} \tag{A.19}$$

we arrive at the hydrodynamical contribution to the RHS of the momentum. Turning our attention to the remaining terms:

$$\begin{aligned}\left.\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t}\right|_{grav} &= -\frac{1}{m_a}\sum_b m_b\left\{\left[\omega W(1-1/W^2)+\frac{1+u}{W}\right]_b\frac{\partial\alpha_b}{\partial x_a^i}\right.\\ &\quad\left.-(\omega WV_j)_b\frac{\partial\beta_b^j}{\partial x_a^i}-\frac{\alpha_b}{2}\left[\omega WV^jV^k+\frac{\sqrt{\gamma}P\gamma^{jk}}{D^\star}\right]_b\frac{\partial\gamma_{jk,b}}{\partial x_a^i}\right\}\delta_{ab}\\ &= -\left[\left(\omega W-\frac{\sqrt{\gamma}P}{D^\star}\right)\frac{\partial\alpha}{\partial x^i}-S_j\frac{\partial\beta^j}{\partial x^i}-\frac{\alpha\sqrt{\gamma}}{2D^\star}\left(\rho_0\omega WV^jV^k+P\gamma^{jk}\right)\frac{\partial\gamma_{jk}}{\partial x^i}\right]_a\\ &= -\left[E\frac{\partial\alpha}{\partial x^i}-S_j\frac{\partial\beta^j}{\partial x^i}-\frac{\alpha}{2}S^{jk}\frac{\partial\gamma_{jk}}{\partial x^i}\right]_a, \end{aligned} \tag{A.20}$$

we see that the gravitational contribution to the RHS of the momentum takes the same form as the continuum equation, 3.60. We now see that the final form of our equation is

$$\begin{aligned}\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t} &= \sum_b m_b\left[\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_a\frac{\partial W_{ab}(h_a)}{\partial x_a^i}+\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b\frac{\partial W_{ab}(h_b)}{\partial x_a^i}\right]\\ &\quad-\left[E\frac{\partial\alpha}{\partial x^i}-S_j\frac{\partial\beta^j}{\partial x^i}-\frac{\alpha}{2}S^{jk}\frac{\partial\gamma_{jk}}{\partial x^i}\right]_a. \end{aligned} \tag{A.21}$$

### A.1.2 The Energy Equation

We will now turn our attention to the equation for the evolution of our energy variable, $\bar{E}$. From Equation 3.48, we rewrite our energy variable as in [61]:

$$\begin{aligned}
\bar{E} &= \alpha E - \beta^i S_i \\
&= \alpha\left(\omega W - \frac{P}{D}\right) - \alpha V^i S_i + v^i S_i \\
&= \alpha\omega W(1 - V^i V_i) - \frac{\alpha P}{W\rho_0} + S_i v^i \\
&= \frac{\alpha}{W}(\omega - \frac{P}{\rho_0}) + S_i v^i \\
&= \frac{\alpha(1+u)}{W} + S_i v^i. \qquad \text{(A.22)}
\end{aligned}$$

We now see that the time derivative of $\bar{E}$ takes the form:

$$\begin{aligned}
\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t} &= \left[S_i\frac{\mathrm{d}v^i}{\mathrm{d}t} + v^i\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1+u}{W}\right)\right]_a \\
&= \left[S_i\frac{\mathrm{d}v^i}{\mathrm{d}t} + v^i\frac{\mathrm{d}S_i}{\mathrm{d}t} + \frac{1+u}{W}\frac{\mathrm{d}\alpha}{\mathrm{d}t} + \frac{\alpha}{W}\frac{\mathrm{d}u}{\mathrm{d}t} - \frac{\alpha(1+u)}{W^2}\frac{\mathrm{d}W}{\mathrm{d}t}\right]_a \qquad \text{(A.23)}
\end{aligned}$$

We will once again use the first law of thermodynamics (Equation A.5) in order to calculate the time derivative of $u$,

$$\begin{aligned}
\frac{\mathrm{d}u_a}{\mathrm{d}t} &= \frac{\partial u_a}{\partial\rho_{0,a}}\frac{\mathrm{d}\rho_{0,a}}{\mathrm{d}t} = \frac{P_a}{\rho_{0,a}^2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{D^\star_a}{\sqrt{\gamma_a}W_a}\right) \\
&= \frac{P_a}{\rho_{0,a}^2}\left[\frac{1}{\sqrt{\gamma}W}\frac{\mathrm{d}D^\star}{\mathrm{d}t} - \frac{D^\star}{2\gamma^{3/2}W}\frac{\mathrm{d}\gamma}{\mathrm{d}t} - \frac{D^\star}{\sqrt{\gamma}W^2}\frac{\mathrm{d}W}{\mathrm{d}t}\right]_a \qquad \text{(A.24)}
\end{aligned}$$

Next, by using the identity [8],

$$\frac{\mathrm{d}\gamma_a}{\mathrm{d}t} = \left[\gamma\gamma^{jk}\frac{\mathrm{d}\gamma_{jk}}{\mathrm{d}t}\right]_a, \qquad \text{(A.25)}$$

we see that the term $(\alpha/W)\mathrm{d}u/\mathrm{d}t$ in Equation A.23 may be written as

$$\left[\frac{\alpha}{W}\frac{\mathrm{d}u}{\mathrm{d}t}\right]_a = \left[\frac{\alpha\sqrt{\gamma}P}{D^{\star 2}}\frac{\mathrm{d}D^\star}{\mathrm{d}t} - \frac{\alpha\sqrt{\gamma}P}{WD^\star}\frac{\mathrm{d}W}{\mathrm{d}t} - \frac{\alpha\sqrt{\gamma}P\gamma^{jk}}{2D^\star}\frac{\mathrm{d}\gamma_{jk}}{\mathrm{d}t}\right]_a. \qquad \text{(A.26)}$$

We will now use Equation A.18 and Equation A.26 in order to split Equation A.23 into terms resulting from hydrodynamic forces and terms resulting from gravitational forces. We see that the hydrodynamic terms take the form:

$$\left.\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t}\right|_{hydro} = v_a^i \left.\frac{\mathrm{d}S_{i,a}}{\mathrm{d}t}\right|_{hydro} + \left[\frac{\alpha\sqrt{\gamma}P}{D^{\star 2}}\frac{\mathrm{d}D^\star}{\mathrm{d}t}\right]_a$$
$$= -\sum_b m_b \left[\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_a v_b^i \cdot \frac{\partial W_{ab}(h_a)}{\partial x_a^i} + \left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b v_a^i \cdot \frac{\partial W_{ab}(h_b)}{\partial x_a^i}\right] \tag{A.27}$$

Turning our attention to the time derivative of the remaining gravitational terms, we will first need the time derivative of the Lorentz factor:

$$\begin{aligned}\frac{\mathrm{d}W_b}{\mathrm{d}t} &= \frac{\mathrm{d}}{\mathrm{d}t}\left[(1-\gamma_{jk}V^jV^k)_b^{-1/2}\right] \\ &= \frac{(1-\gamma_{jk}V^jV^k)_b^{-3/2}}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(\gamma_{jk}V^jV^k\right)_b \\ &= \frac{W_b^3}{2}\left[V_b^jV_b^k\frac{\mathrm{d}\gamma_{jk,b}}{\mathrm{d}t} + (2\gamma_{jk}V^j)_b\frac{\mathrm{d}V_b^k}{\mathrm{d}t}\right] \\ &= W_b^3\left[\frac{V_b^jV_b^k}{2}\frac{\mathrm{d}\gamma_{jk,b}}{\mathrm{d}t} - \left(\frac{V_kV^k}{\alpha}\right)_b\frac{\mathrm{d}\alpha_b}{\mathrm{d}t} + \left(\frac{V_j}{\alpha}\right)_b\frac{\mathrm{d}v_b^j}{\mathrm{d}t} + \left(\frac{V_j}{\alpha}\right)_b\frac{\mathrm{d}\beta_b^j}{\mathrm{d}t}\right].\end{aligned} \tag{A.28}$$

By plugging Equation A.28 into the remaining terms from Equation A.23 and collecting like terms, we see that:

$$\left.\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t}\right|_{grav} = \left[E\frac{\partial\alpha}{\mathrm{d}t} - S_j\frac{\partial\beta^j}{\mathrm{d}t} - \frac{\alpha}{2}S^{jk}\frac{\partial\gamma_{jk}}{\mathrm{d}t}\right]_a. \tag{A.29}$$

We now write the complete time evolution equation for the energy as:

$$\begin{aligned}\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t} &= \left.\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t}\right|_{hydro} + \left.\frac{\mathrm{d}\bar{E}_a}{\mathrm{d}t}\right|_{grav} \\ &= -\sum_b m_b\left[\left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_a v_b^i\cdot\frac{\partial W_{ab}(h_a)}{\partial x_a^i} + \left(\frac{\alpha\sqrt{\gamma}P}{\Omega D^{\star 2}}\right)_b v_a^i\cdot\frac{\partial W_{ab}(h_b)}{\partial x_a^i}\right] \\ &+ \left[E\frac{\partial\alpha}{\mathrm{d}t} - S_j\frac{\partial\beta^j}{\mathrm{d}t} - \frac{\alpha}{2}S^{jk}\frac{\partial\gamma_{jk}}{\mathrm{d}t}\right]_a.\end{aligned} \tag{A.30}$$

# REFERENCES


[1] Abbott, B., Abbott, R., Abbott, T., Abernathy, M., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R., and others, "Gw151226: Observation of gravitational waves from a 22-solar-mass binary black hole coalescence," *Physical Review Letters*, vol. 116, no. 24, p. 241103, 2016.

[2] Abbott, B., Abbott, R., Abbott, T., Abernathy, M., Acernese, F., Ackley, K., Adams, C., Adams, T., Addesso, P., Adhikari, R., and others, "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, p. 061102, 2016.

[3] Agertz, O., Moore, B., Stadel, J., Potter, D., Miniati, F., Read, J., Mayer, L., Gawryszczak, A., Kravtsov, A., Nordlund, Å., and others, "Fundamental differences between sph and grid methods," *Monthly Notices of the Royal Astronomical Society*, vol. 380, no. 3, pp. 963–978, 2007.

[4] Allen, G., Benger, W., Goodale, T., Hege, H.-C., Lanfermann, G., Merzky, A., Radke, T., Seidel, E., and Shalf, J., "The cactus code: A problem solving environment for the grid," in *High-Performance Distributed Computing, 2000. Proceedings. The Ninth International Symposium on*, pp. 253–260, IEEE, 2000.

[5] Amdahl, G. M., "Validity of the single processor approach to achieving large scale computing capabilities," in *Proceedings of the April 18-20, 1967, Spring Joint Computer Conference*, pp. 483–485, ACM, 1967.

[6] Arnowitt, R., Deser, S., and Misner, C. W., "Dynamical structure and definition of energy in general relativity," *Physical Review*, vol. 116, no. 5, p. 1322, 1959.

[7] Banyuls, F., Miralles, J. A., and others, "Numerical {3+ 1} general relativistic hydrodynamics: A local characteristic approach," *The Astrophysical Journal*, vol. 476, no. 1, p. 221, 1997.

[8] Baumgarte, T. W. and Shapiro, S. L., *Numerical Relativity: Solving Einstein's Equations on the Computer*. Cambridge University Press, 2010.

[9] Bender, J., Duriez, C., Jaillet, F., and Zachmann, G., "A parallel architecture for iisph fluids,"

[10] Berger, M. J. and Oliger, J., "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of computational Physics*, vol. 53, no. 3, pp. 484–512, 1984.

[11] Bethe, H., Fuchs, K., Hirschfelder, J., Magee, J., Peieris, R., and von Neumann, J., "Blast wave," *DTIC Document*, 1947.

[12] Bondi, H., "On spherically symmetrical accretion," *Monthly Notices of the Royal Astronomical Society*, vol. 112, no. 2, pp. 195–204, 1952.

[13] Bonr, J., "Latency numbers every programmer should know." https://gist.github.com/jboner/2841832, 2012.

[14] Brookshaw, L., "A method of calculating radiative heat diffusion in particle simulations," in *Proceedings of the Astronomical Society of Australia*, vol. 6, pp. 207–210, 1985.

[15] Cabezón, R. M., Garcia-Senz, D., and Escartín, J. A., "Testing the concept of integral approach to derivatives within the smoothed particle hydrodynamics technique in astrophysical scenarios," *Astronomy & Astrophysics*, vol. 545, p. A112, 2012.

[16] Carroll, S. M., *Spacetime and geometry. An introduction to general relativity*, vol. 1. 2004.

[17] Centrella, J. and Wilson, J. R., "Planar numerical cosmology. ii-the difference equations and numerical tests," *The Astrophysical Journal Supplement Series*, vol. 54, pp. 229–249, 1984.

[18] Cossins, P. J., *Smoothed Particle Hydrodynamics*. PhD thesis, 2010.

[19] Courant, R., Friedrichs, K., and Lewy, H., "Über die partiellen differenzengleichungen der mathematischen physik," *Mathematische annalen*, vol. 100, no. 1, pp. 32–74, 1928.

[20] Cullen, L. and Dehnen, W., "Inviscid smoothed particle hydrodynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 408, no. 2, pp. 669–683, 2010.

[21] Dehnen, W. and Aly, H., "Improving convergence in smoothed particle hydrodynamics simulations without pairing instability," *Monthly Notices of the Royal Astronomical Society*, vol. 425, no. 2, pp. 1068–1082, 2012.

[22] Dubinski, J., "A parallel tree code," *New Astronomy*, vol. 1, no. 2, pp. 133–147, 1996.

[23] Eitz, M. and Lixu, G., "Hierarchical spatial hashing for real-time collision detection," in *Shape Modeling and Applications, 2007. SMI'07. IEEE International Conference on*, pp. 61–70, IEEE, 2007.

[24] Faber, J. A., Grandclément, P., and Rasio, F. A., "Mergers of irrotational neutron star binaries in conformally flat gravity," *Physical Review D*, vol. 69, no. 12, p. 124036, 2004.

[25] Fock, V., *The Theory of Space, Time and Gravitation.* Pergamon Press book, Pergamon Press; [distributed in the Western Hempisphere by Macmillan, New York], 1964.

[26] Font, J. A., "Numerical hydrodynamics and magnetohydrodynamics in general relativity," *Living Rev. Relativity*, vol. 11, no. 7, p. 3, 2008.

[27] Garcia-Senz, D., Cabezón, R. M., and Escartín, J. A., "Improving smoothed particle hydrodynamics with an integral approach to calculating gradients," *Astronomy & Astrophysics*, vol. 538, p. A9, 2012.

[28] Gingold, R. A. and Monaghan, J. J., "Smoothed particle hydrodynamics: theory and application to non-spherical stars," *Monthly notices of the royal astronomical society*, vol. 181, no. 3, pp. 375–389, 1977.

[29] Gottlieb, S. and Shu, C.-W., "Total variation diminishing runge-kutta schemes," *Mathematics of computation of the American Mathematical Society*, vol. 67, no. 221, pp. 73–85, 1998.

[30] Gresho, P. M., "On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. part 1: theory," *International Journal for Numerical Methods in Fluids*, vol. 11, no. 5, pp. 587–620, 1990.

[31] Harten, A., "High resolution schemes for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 49, no. 3, pp. 357–393, 1983.

[32] Helmholtz, P., "Xliii. on discontinuous movements of fluids," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 36, no. 244, pp. 337–346, 1868.

[33] Herrmann, F., Hinder, I., Shoemaker, D., and Laguna, P., "Unequal mass binary black hole plunges and gravitational recoil," *Classical and Quantum Gravity*, vol. 24, no. 12, p. S33, 2007.

[34] Hinder, I., Vaishnav, B., Herrmann, F., Shoemaker, D. M., and Laguna, P., "Circularization and final spin in eccentric binary-black-hole inspirals," *Physical Review D*, vol. 77, no. 8, p. 081502, 2008.

[35] Hobbs, A., Power, C., Nayakshin, S., and King, A. R., "On the modelling of accretion onto smbhs," 2011.

[36] Hugoniot, H., "Sur la propagation du movement dans les corps et spécialement dans les gaz parfaits," *J. Ecole Polytechnique*, vol. 58, pp. 1–125, 1889.

[37] Laguna, P., Miller, W. A., and Zurek, W. H., "Smoothed particle hydrodynamics near a black hole," *The Astrophysical Journal*, vol. 404, pp. 678–685, 1993.

[38] Landau, L. and Lifshitz, E., "The classical theory of fields: Volume 2 (course of theoretical physics series)," 2000.

[39] Landau, L. and Lifshits, E., *Fluid mechanics*. A-W series in advanced physics, Pergamon Press, 1959.

[40] Landshoff, R., "A numerical method for treating fluid flow in the presence of shocks," *DTIC Document*, 1955.

[41] Löffler, F., Faber, J., Bentivegna, E., Bode, T., Diener, P., Haas, R., Hinder, I., Mundim, B. C., Ott, C. D., Schnetter, E., and others, "The einstein toolkit: a community computational infrastructure for relativistic astrophysics," *Classical and Quantum Gravity*, vol. 29, no. 11, p. 115001, 2012.

[42] Lucy, L. B., "A numerical approach to the testing of the fission hypothesis," *The astronomical journal*, vol. 82, pp. 1013–1024, 1977.

[43] Martí, J. M., Ibáñez, J. M., and Miralles, J. A., "Numerical relativistic hydrodynamics: local characteristic approach," *Physical Review D*, vol. 43, no. 12, p. 3794, 1991.

[44] Michel, F. C., "Accretion of matter by condensed objects," *Astrophysics and Space Science*, vol. 15, no. 1, pp. 153–160, 1972.

[45] Misner, C., Thorne, K., and Wheeler, J., *Gravitation*. No. pt. 3 in Gravitation, W. H. Freeman, 1973.

[46] Monaghan, J. and Gingold, R., "Shock simulation by the particle method sph," *Journal of Computational Physics*, vol. 52, no. 2, pp. 374–389, 1983.

[47] Monaghan, J. and Price, D., "Variational principles for relativistic smoothed particle hydrodynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 328, no. 2, pp. 381–392, 2001.

[48] Montero, P. J., Baumgarte, T. W., and Müller, E., "General relativistic hydrodynamics in curvilinear coordinates," *Physical Review D*, vol. 89, no. 8, p. 084043, 2014.

[49] Moore, D., "Fast hilbert curve generation, sorting, and range queries," *Internet: http://web.archive.org/web/20050212162158/http://www.caam.rice.edu/~dougm/twiddle/Hilbert/*, 1999.

[50] Morris, J. and Monaghan, J., "A switch to reduce sph viscosity," *Journal of Computational Physics*, vol. 136, no. 1, pp. 41–50, 1997.

[51] Nelson, R. P. and Papaloizou, J. C., "Variable smoothing lengths and energy conservation in smoothed particle hydrodynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 270, no. 1, pp. 1–20, 1994.

[52] Philipp, M., Mundim, B. C., Faber, J. A., Haas, R., Noble, S. C., Bode, T., Frank, L., Ott, C. D., Reisswig, C., Schnetter, E., and others, "Grhydro: a new open-source general-relativistic magnetohydrodynamics code for the einstein toolkit," *Classical and quantum gravity*, vol. 31, no. 1, p. 015005, 2014.

[53] Price, D. J., "Modelling discontinuities and kelvin–helmholtz instabilities in sph," *Journal of Computational Physics*, vol. 227, no. 24, pp. 10040–10057, 2008.

[54] Rankine, W. M., "On the thermodynamic theory of waves of finite longitudinal disturbance," *Philosophical Transactions of the Royal Society of London*, pp. 277–288, 1870.

[55] Rantsiou, E., Kobayashi, S., Laguna, P., and Rasio, F. A., "Mergers of black hole-neutron star binaries. i. methods and first results," *The Astrophysical Journal*, vol. 680, no. 2, p. 1326, 2008.

[56] Read, J., Hayfield, T., and Agertz, O., "Resolving mixing in smoothed particle hydrodynamics," *Monthly Notices of the Royal Astronomical Society*, vol. 405, no. 3, pp. 1513–1530, 2010.

[57] Robertson, B. E., Kravtsov, A. V., Gnedin, N. Y., Abel, T., and Rudd, D. H., "Computational eulerian hydrodynamics and galilean invariance," *Monthly Notices of the Royal Astronomical Society*, vol. 401, no. 4, pp. 2463–2476, 2010.

[58] Rosswog, S., "Boosting the accuracy of sph techniques: Newtonian and special-relativistic tests," *Monthly Notices of the Royal Astronomical Society*, vol. 448, no. 4, pp. 3628–3664, 2015.

[59] Rosswog, S., "Astrophysical smooth particle hydrodynamics," *New Astronomy Reviews*, vol. 53, no. 4, pp. 78–104, 2009.

[60] Rosswog, S., "Conservative, special-relativistic smoothed particle hydrodynamics," *Journal of Computational Physics*, vol. 229, no. 22, pp. 8591–8612, 2010.

[61] Rosswog, S., "Relativistic smooth particle hydrodynamics on a given background spacetime," *Classical and Quantum Gravity*, vol. 27, no. 11, p. 114108, 2010.

[62] Rosswog, S., "Sph methods in the modelling of compact objects," *arXiv preprint arXiv:1406.4224*, 2014.

[63] Schnetter, E., Hawley, S. H., and Hawke, I., "Evolutions in 3d numerical relativity using fixed mesh refinement," *Classical and quantum gravity*, vol. 21, no. 6, p. 1465, 2004.

[64] Schoenberg, I. J., "Contributions to the problem of approximation of equidistant data by analytic functions: Part bon the problem of osculatory interpolation. a second class of analytic approximation formulae," *Quarterly of Applied Mathematics*, vol. 4, no. 2, pp. 112–141, 1946.

[65] Schornbaum, F., "Hierarchical hash grids for coarse collision detection," *Student Thesis, University of Erlangen-Nuremberg*, 2009.

[66] Siegler, S. and Riffert, H., "Smoothed particle hydrodynamics simulations of ultrarelativistic shocks with artificial viscosity," *The Astrophysical Journal*, vol. 531, no. 2, p. 1053, 2000.

[67] Sod, G. A., "A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws," *Journal of computational physics*, vol. 27, no. 1, pp. 1–31, 1978.

[68] Springel, V., "The cosmological simulation code gadget-2," *Monthly Notices of the Royal Astronomical Society*, vol. 364, no. 4, pp. 1105–1134, 2005.

[69] Springel, V., "Smoothed particle hydrodynamics in astrophysics," *Annual Review of Astronomy and Astrophysics*, vol. 48, pp. 391–430, 2010.

[70] Springel, V. and Hernquist, L., "Cosmological smoothed particle hydrodynamics simulations: the entropy equation," *Monthly Notices of the Royal Astronomical Society*, vol. 333, no. 3, pp. 649–664, 2002.

[71] Taylor, G., "The formation of a blast wave by a very intense explosion. i. theoretical discussion," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, pp. 159–174, 1950.

[72] Teschner, M., Heidelberger, B., Müller, M., Pomerantes, D., and Gross, M. H., "Optimized spatial hashing for collision detection of deformable objects.," in *VMV*, vol. 3, pp. 47–54, 2003.

[73] Thacker, R., Tittley, E., Pearce, F., Couchman, H., and Thomas, P., "Smoothed particle hydrodynamics in cosmology: a comparative study of implementations," *Monthly Notices of the Royal Astronomical Society*, vol. 319, no. 2, pp. 619–648, 2000.

[74] Thompson, K. W., "The special relativistic shock tube," *Journal of Fluid Mechanics*, vol. 171, pp. 365–375, 1986.

[75] Thomson, W., "Xlvi. hydrokinetic solutions and observations," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 42, no. 281, pp. 362–377, 1871.

[76] VonNeumann, J. and Richtmyer, R. D., "A method for the numerical calculation of hydrodynamic shocks," *Journal of applied physics*, vol. 21, no. 3, pp. 232–237, 1950.

[77] Wendland, H., "Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree," *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.